



TESIS - TE142599

**KLASIFIKASI *IMAGE SEQUENCE* HASIL
PENGELASAN MENGGUNAKAN METODE
CONVOLUTION NEURAL NETWORK (CNN) UNTUK
NON DESTRUCTIVE TEST.**

AGUS KHUMAI
2215205204

DOSEN PEMBIMBING
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Dr. Eko Mulyanto Yuniarno, ST., MT.

PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



TESIS - TE142599

**KLASIFIKASI *IMAGE SEQUENCE* HASIL
PENGELASAN MENGGUNAKAN METODE
CONVOLUTION NEURAL NETWORK (CNN) UNTUK
NON DESTRUCTIVE TEST.**

AGUS KHUMAI
2215205204

DOSEN PEMBIMBING
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
Dr. Eko Mulyanto Yuniarno, ST., MT.

PROGRAM MAGISTER
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

oleh:

Agus Khumaidi
NRP. 2215205204

Tanggal Ujian : 15 Juni 2017
Periode Wisuda : September 2017

Disetujui oleh:

1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng. (Pembimbing I)
NIP: 195809161986011001
2. Dr. Eko Mulyanto Yuniarno, ST., MT. (Pembimbing II)
NIP: 196806011995121009
3. Dr. Supeno Mardi Susiki Nugroho, ST., MT. (Penguji)
NIP: 197003131995121001
4. Dr. Surya Sumpeno, ST., M.Sc. (Penguji)
NIP: 196906131997021003
5. Dr. Diah Puspito Wulandari, ST., M.Sc. (Penguji)
NIP: 198012192005012001



Dekan Fakultas Teknologi Elektro

Dr. Tri Arief Sardjono, S.T., M.T.
NIP. 197002121995121001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul **“KLASIFIKASI *IMAGE SEQUENCE* HASIL PENGELASAN MENGGUNAKAN METODE CONVOLUTION NEURAL NETWORK (CNN) UNTUK *NON DESTRUCTIVE TEST*”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2017



Agus Khumaidi

NRP. 2215205204

Halaman ini sengaja dikosongkan

KLASIFIKASI *IMAGE SEQUENCE* HASIL PENGELASAN MENGUNAKAN METODE CONVOLUTION NEURAL NETWORK (CNN) UNTUK *NON DESTRUCTIVE TEST*

Nama mahasiswa : Agus Khumaidi
NRP : 2215205204
Pembimbing : 1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
2. Dr. Eko Mulyanto, ST., M.T.

ABSTRAK

Visual inspection adalah salah satu metode *Non Destructive Test (NDT)* yang sangat penting untuk proses uji hasil kualitas pengelasan, *visual inspection* dilakukan sebagai seleksi tahap awal sebelum hasil pengelasan dilanjutkan ke proses *Destructive Test (DT)* (Jurandir Primo, 2012). Proses *visual inspection* ini masih menggunakan cara manual yaitu dengan menggunakan pengelihatan manusia, sehingga hasil pengujian masih sangat subjektif.

Pada penelitian ini proses *visual inspection* akan dilakukan melalui pengolahan citra pada *image sequence* dengan menggunakan metode Convolution Neural Network (CNN) untuk klasifikasi cacat pada hasil pengelasan. Terdapat 4 vektor output klasifikasi yaitu *good*, *over spatter*, *porosity*, dan *undercut*. Arsitektur CNN yang digunakan memiliki 10 *feature map*, 1 layer tersembunyi dengan 64 hidden Neuron, jumlah iterasi 60 dan laju pembelajaran α yaitu 0,2. Hasil akurasi menunjukkan CNN mampu mengklasifikasi kategori cacat las dengan tingkat akurasi untuk data validasi bernilai 70,85% dari total 24 data yang diujikan.

Kata kunci: *Visual Inspection*, Non Destructive Test, Convolution Neural Network, Cacat Pengelasan.

Halaman ini sengaja dikosongkan

WELDING DEFECT CLASSIFICATION ON IMAGE SEQUENCE USING CONVOLUTION NEURAL NETWORK (CNN) FOR NON DISTRUCTIVE TEST

By : Agus Khumaidi
Student Identity Number : 2215205204
Supervisor(s) : 1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.
2. Dr. Eko Mulyanto, ST., M.T.

ABSTRACT

Visual inspection is one of the most important Non Destructive Test (NDT) methods to test the results of weld quality. This inspection method is performed as initial selection before Destructive Test (DT) process (Jurandir Primo, 2012). Nowadays, visual inspection process is still using the manual way by using human vision, so the test results are still very subjective.

In this research, visual inspection process will be done through image processing in image sequence by using method of convolution neural network (CNN) to classify the result of welding defect. There are 4 vectors of classification output, these are good, over spatter, porosity, and undercut. The CNN architecture used has 10 feature maps, 1 hidden layer with 64 hidden Neurons, 60 numbers of iterations and learning rate α of 0.2. The results show that CNN is able to classify the welding defects with the accuracy rate of 70.85% for the validation data from the total of 24 data tested.

Key words: *Visual Inspection*, Non Destructive Test, Convolution Neural Network, Welding Defect.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Syukur Alhamdulillah penulis panjatkan kepada Allah SWT, berkat rahmat dan karunia-Nya tesis ini dapat diselesaikan. Penulis menyampaikan ucapan terima kasih kepada semua pihak yang telah membantu dalam menyelesaikan tesis ini. Secara khusus penulis menyampaikan ucapan terima kasih yang tulus kepada :

1. Kedua orang tua penulis Sujak dan Nur Azizah juga kakak Mahit Masrufi, dan Adik Taufiq Rahman yang telah memberikan dukungan sepenuhnya kepada penulis selama studi magister.
2. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng. dan Dr. Eko Mulyanto Yuniarno., S.T., M.T. selaku dosen pembimbing yang telah mengarahkan, memberi koreksi, memberi motivasi dan mendukung sepenuhnya dalam pembuatan tesis ini.
3. Dr. Eko Mulyanto Yuniarno., S.T., M.T. selaku koordinator bidang keahlian Jaringan Cerdas Multimedia.
4. Dewan penguji yang telah memberikan masukan dan saran dalam tesis ini.
5. Beasiswa Unggulan yang telah memberikan dukungan biaya studi magister ongoing.
6. Politeknik Perkapalan Negeri Surabaya dan jurusan Teknik Kelistrikan Kapal yang telah memberikan kesempatan kepada saya untuk dapat melanjutkan studi magister.
7. Sahabat disekeliling saya yang telah membantu dalam penyelesaian penelitian ini, terutama kakak saya Riyan Yudha Aditya.
8. Semua pihak yang telah membantu dalam proses pengerjaan tesis ini. Penulis menyadari bahwa tesis ini masih jauh dari sempurna. Oleh karenanya masukan, saran dan kritik untuk perbaikan sangat diharapkan. Besar harapan tesis ini dapat bermanfaat bagi penelitian-penelitian selanjutnya.

Surabaya, 13 Juli 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	vi
PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan	3
1.4 Manfaat	3
1.5 Batasan Masalah	3
BAB 2 KAJIAN PUSTAKA	5
2.1 Cacat Pengelasan (Weld Defects Inspection)	5
2.1.1 Incoplete Fusion (Cold Lap)	5
2.1.2 Lack of Penetration	6
2.1.3 Porosity	7
2.1.4 Undercutting	8
2.1.5 Craters	8
2.1.6 Over Spatter	9
2.2 Model Warna RGB	10
2.3 Model Warna Grayscale	10
2.4 Neural Network (NN)	11

2.4.1 Fungsi Aktivasi	14
2.4.2 Algoritma Backpropagation.....	16
2.5 Convolution Neural Network (CNN)	18
BAB 3 METODOLOGI PENELITIAN	25
3.1 Metode Usulan.....	28
3.2 Tahapan Pelatihan Dataset.....	29
3.3 Tahapan Pengujian Data	32
3.4 Konversi Data RGB to Gray	33
3.5 Proses Konvolusi	34
3.6 Proses ReLu dan Maxpooling.....	37
3.7 Fully Connected.....	38
BAB 4 HASIL DAN PEMBAHASAN	41
4.1 Pembentukan Dataset.....	41
4.2 Konvolusi Citra.....	42
4.3 Hasil ReLu dan Maxpooling Citra.....	44
4.4 Perhitungan Bobot	46
4.4.1 Analisa Jumlah Neuron Pada Hidden Layer.....	46
4.4.2 Analisa Arsitektur CNN 1 dan CNN2.	48
4.5 Proses Training dan Validasi	52
4.6 Pengujian Real Time Untuk Hasil Pengelasan	61
BAB 5 PENUTUP	69
5.1 Kesimpulan	69
5.2 Saran	69
DAFTAR PUSTAKA.....	71
BIODATA PENULIS	73

TABLE OF CONTENTS

VALIDITY SHEET	vi
STATEMENT OF ORIGINALITY	v
ABSTRACT IN INDONESIA.....	vii
ABSTRACT.....	ix
PREFACE	xi
TABLE OF CONTENTS.....	xiii
LIST OF FIGURES.....	xvii
LIST OF TABLES	xix
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Statement of The Problem.....	3
1.3 Purpose	3
1.4 Benefits.....	3
1.5 Scope and Limit of Study.....	3
CHAPTER II REVIEW OF RELATED LITERATURE	5
2.1 Welding Defects (Weld Defects Inspection).....	5
2.1.1 Incoplete Fusion (Cold Lap)	5
2.1.2 Lack of Penetration	6
2.1.3 Porosity.....	7
2.1.4 Undercutting.....	8
2.1.5 Craters	8
2.1.6 Over Spatter.....	9
2.2 RGB Color Model	10
2.3 Grayscale Color Model	10
2.4 Neural Network (NN).....	11
2.4.1 Activation Function.....	14
2.4.2 Backpropagation Algorithm	16

2.5	Convolution Neural Network (CNN)	18
CHAPTER 3 RESEARCH METHODOLOGY		25
3.1	Proposed Method	28
3.2	Stages of Dataset Training	29
3.3	Stages of Data Testing	32
3.4	Converting RGB to Gray Data	33
3.5	The Process of convolution	34
3.6	ReLu and Maxpooling Processes	37
3.7	Fully Connected	38
CHAPTER 4 RESULTS AND DISCUSSION		41
4.1	Dataset Formation	41
4.2	Image Convolution	42
4.3	Results of ReLu and Maxpooling Image	44
4.4	Reforestation Weights	46
4.4.1	Analysis of Number of Neurons in Hidden Layer	46
4.4.2	CNN 1 and CNN2 Architectural Analysis	48
4.5	Training and Validation Process	52
4.6	Real Time Testing For Welding Results	61
CHAPTER 5 CLOSING		69
5.1	Conclusions	69
5.2	Suggestions	69
REFERENCE		71
BIBLIOGRAPHY		73

DAFTAR GAMBAR

Gambar 2.1 Bentuk sambungan las (fusion), (a) Good Fusion, (b) Poor Fussion , (c) Bridging Fusion, (d) Double Bevel Groove Welding With Incomplete Fussion, (e) Incomplete Fusion In A Grooved Butt Joint, (f) Incomplete Fusion One Log Of A Butt Joint.....	6
Gambar 2.2 Jenis kegagalan penetrasi, (a) Excessive Penetration, (b) Good Penetration, (c) Lack of Penetration.....	7
Gambar 2.3 Welding porosity	7
Gambar 2.4 Welding undercut	8
Gambar 2.5 Retakan pada pengelasan	9
Gambar 2.6 Over spatter	9
Gambar 2.7 Model Warna RGB.....	10
Gambar 2.8 Hasil Konversi Ruang Warna RGB ke Grayscale.....	11
Gambar 2.9 Syaraf secara biologi.	12
Gambar 2.10 Struktur MLP.....	13
Gambar 2.11 Fungsi step.....	15
Gambar 2.12 Fungsi ramp.....	15
Gambar 2.13 Fungsi sigmoid unipolar.....	15
Gambar 2.14 Fungsi sigmoid bipolar.....	16
Gambar 2. 15 Citra Hasil Konvolusi, (a) Citra Asli, (b) Blurring, (c) Embossing, (d) Sharpening, (e) Edge.	20
Gambar 2.16 Operasi Konvolusi.....	21
Gambar 2.17 Operasi Max Pooling.....	22
Gambar 3.1 Metode Penelitian.....	25
Gambar 3.2 Desain Arsitektur CNN 1 (Stanford, 2016).....	26
Gambar 3.3 Desain Arsitektur CNN 2	27
Gambar 3.4 Contoh citra hasil pengelasan baik (a) dan cacat las (b).	30
Gambar 3.5 Alur proses pelatihan.....	31
Gambar 3.6 Tahapan Pengujian.	32
Gambar 3.7 Grafik Perubahan Nilai Gaussian 2D Terhadap Koordinat X dan Y	35
Gambar 3.8 Konfigurasi Persamaan Gaussian Dengan Koordinat X dan Y.....	35

Gambar 3.9 Citra Baru Hasil Konvolusi	36
Gambar 3.10 Citra Baru Hasil Proses ReLu dan Maxpooling	37
Gambar 4.1 Hasil Pengelasan Kondisi Good (GO) , Over Spatter (OS), Porosity (PO), dan Undercut (UC).....	41
Gambar 4.2 Jaringan Neural Network yang Terbentuk, (a) Arsitektur CNN 1, (b) Arsitektur CNN 2.....	48
Gambar 4.3 Hasil MSE Untuk Data Training (Biru) Dan Data Testing (Hijau), (a) Arsitektur 1, (b) Arsitektur 2.	55
Gambar 4.4 Hasil Akurasi Untuk Data Training (Biru) Dan Data Testing (Hijau) , (a) Arsitektur 1, (b) Arsitektur 2.....	56
Gambar 4.5 Contoh hasil pengelasan yang akan diuji. GO (Good), UC(Undercut), OS (Overspater), PO (Porosity).....	62

DAFTAR TABEL

Tabel 4.1 Feature Maps untuk Dimensi Kernel 5 x 5 dengan Gaussian Feature pada Citra Pengelasan	42
Tabel 4.2 Hasil ReLu dan Maxpooling pada Feature Maps.....	44
Tabel 4.3 Korelasi Modifikasi Jumlah Neuron Terhadap Perubahan Akurasi	47
Tabel 4.4 Bobot Training Input Layer dan Hidden Layer untuk Arsitektur CNN 1.	49
Tabel 4.5 Bobot Training Input Layer dan Hidden Layer untuk Arsitektur CNN 2.	50
Tabel 4.6 Bobot Training Hidden Layer dan Output Layer untuk Arsitektur CNN 1 dan 2.....	51
Tabel 4.7 Parameter Input Dataset	52
Tabel 4.8 Nilai Biner Vektor Output untuk Masing – Masing Kategori	53
Tabel 4. 9 Respon Akurasi Data Validasi	58
Tabel 4.10 Hasil Prediksi Untuk 24 Data Validasi	60
Tabel 4.11 Distribusi sebaran data citra.....	61
Tabel 4.12 Hasil Klasifikasi Pengelasan yang Baik Secara Real Time.	62
Tabel 4.13 Hasil Klasifikasi Pengelasan dengan Cacat Over Spatter.....	64
Tabel 4.14 Hasil Klasifikasi Pengelasan dengan Cacat Porosity.	65
.Tabel 4.15 Hasil Klasifikasi Pengelasan dengan Cacat Undercut	66
Tabel 4.16 Persentase Keberhasilan Pengujian.....	67

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Metode inspeksi hasil pengelasan dilakukan melalui 2 metode yaitu *Non Destructive Testing (NDT)* dan *Destructive Testing (DT)* (Jurandir Primo, 2012). NDT adalah aktivitas tes atau inspeksi terhadap suatu benda untuk mengetahui adanya cacat, retak, *discontinuity*, dan lain tanpa merusak benda yang kita tes atau inspeksi. Pada dasarnya, tes ini dilakukan untuk menjamin bahwa material yang kita gunakan masih aman dan belum melewati *damage tolerance*. Metode NDT dibagi menjadi beberapa kategori yaitu *Visual Inspection (VT)*, *Liquid Penetrant Testing (LT)*, *Magnetic Particle Testing (MT)*, *Radiographic Testing (RT)*, *Ultrasonic Testing (UT)*, *Eddy Current Testing (ET)*, *Acoustic Emission Testing (AET)*, *Phased Array Ultrasonics (PA)*, *Time Of Flight Diffraction (TOFD)*.

Destructive Testing (DT) adalah suatu cara pengujian hasil pengelasan dengan cara merusak hasil pengelasan yang diuji. Tujuannya adalah untuk mengetahui kekuatan las terhadap suatu jenis pembebanan. Pengujian DT antara lain (Jurandir Primo, 2012) yaitu pengujian tarik, pengujian beban kejut (*impact*), pengujian kekerasan, pengujian macro (*structure test*).

Klasifikasi citra dengan menggunakan CNN pertama kali dilakukan, pada tahun 1989 oleh Yann LeCun dan timnya, saat itu mereka berhasil melakukan klasifikasi citra pada kode zip menggunakan metode *feedforward* Neural Network yang kemudian dikenal dengan nama Convolution Neural Network (Y. Le Cun, 1990). Dengan berkembang perangkat Graphical Processing Unit (GPU) metode Deep Learning Neural Network (DNN) semakin dikembangkan dengan akurasi pengenalan yang menyaingi kemampuan manusia pada dataset tertentu (I Wayan Suartika E. P, 2016).

Pengujian hasil pengelasan dengan menggunakan kecerdasan telah dilakukan oleh J. Mirapeix tahun 2007, pada penelitiannya bertujuan untuk mengetahui cacat pada pengelasan dengan menggunakan sinyal plasma spectrum dari Spectrometer yang kemudian diekstraksi dengan menggunakan Principal Component Analitic (PCA) dan klasifikasi menggunakan Artificial Neural Network untuk klasifikasi tingkat panas dari electrode (J. Mirapeix, 2007).

Jayenda Kumar tahun 2014 pada penelitiannya, citra yang digunakan adalah citra hasil radio graphic dengan ekstraksi fitur menggunakan metode Gray Level Co-occurrence Matrix (GLCM) dan untuk proses *training* menggunakan Back Propagation Neural Network (BPNN) (Jayendra Kumar, 2014). Karena data yang digunakan adalah hasil dari radio graphic maka *noise* yang dimiliki sangat kecil sehingga sangat mudah dalam mengambil *feature*, vektor input dari ekstraksi GLCM yang digunakan sejumlah 4 unit yang terdiri dari contrast, correlation, energy, dan homogeneity. Karena vektor input yang digunakan hanya 4 unit maka bisa dipastikan nilai output setiap vektor akan sangat berdekatan, bukan hanya itu untuk mendapatkan dataset yang baik maka image yang digunakan harus memiliki *texture* yang mencolok atau berbeda untuk setiap kelasnya, tetapi akan kesulitan apabila dataset mempunyai bentuk pola yang hampir sama.

Dari rumusan diatas, peneliti mengajukan metode usulan untuk melakukan klasifikasi hasil pengelasan yaitu dengan menggunakan deep learning dengan metode Convolution Neural Network (CNN), input citra yang digunakan pada penelitian ini adalah citra yang diambil secara langsung melalui kamera sehingga didapatkan klasifikasi secara *real time*. Sebelum melalui proses testing dan training input citra dari kamera yang masih memiliki ruang warna RGB akan diubah menjadi ruang warna grayscale. Sedangkan untuk dataset tidak perlu dilakukan ekstraksi *feature* satu persatu, cukup dengan melakukan pengelompokan atau *labeling*. Ekstraksi citra akan dilakukan secara otomatis dengan menggunakan konvolusi citra dengan Gaussian kernel. Secara konseptual CNN terdiri atas 3 layer yaitu convolution layer, subsampling layer, dan fully connected layer.

1.2 Rumusan Masalah

Visual inspection mengharuskan pengecekan dilakukan dengan tepat dan dapat dilakukan secara *real time* karena kemungkinan hasil pengelasan yang harus diinspeksi dalam jumlah banyak. Dari permasalahan tersebut diharapkan dengan CNN dapat melakukan klasifikasi yang baik meskipun dengan menggunakan citra asli.

1.3 Tujuan

Tujuan dari penelitian ini adalah mendapatkan arsitektur CNN yang tepat dan optimal untuk proses visual inspection sehingga didapatkan hasil klasifikasi yang baik sehingga memiliki nilai error yang kecil.

1.4 Manfaat

Adapun manfaat kedepan dari penelitian ini adalah

1. Meningkatkan akurasi dan efisiensi proses inspeksi visual hasil pengelasan.
2. Dapat diterapkan dalam proses pengelasan otomatis.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah

1. Klasifikasi hasil pengelasan yang digunakan sebagai acuan adalah berdasarkan good, cacat *porosity*, cacat *over spatter*, dan cacat *undercutting*.
2. Tipe pengelasan yang diambil adalah hasil pengelesan yang menggunakan probe atau electrode (SMAW).

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA

2.1 Cacat Pengelasan (*Weld Defects Inspection*)

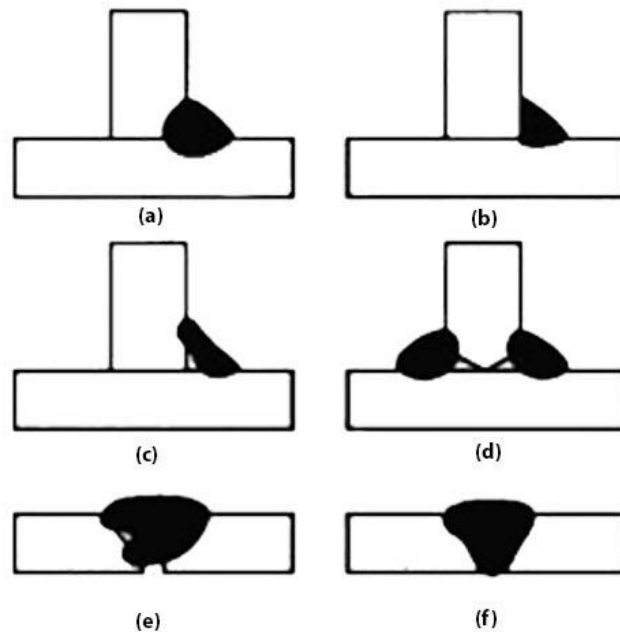
Hal yang sangat penting untuk menjadi pertimbangan dalam proses inspeksi las terdiri atas dua bagian yang berbeda yaitu cacat las (*weld defect*) dan sambungan las (*weld discontinue*) (Jurandir Primo, 2012). Ketika dalam satu bidang pengelasan terdapat satu atau lebih bagian las yang tidak tersambung (*discontinue*) maka bisa disebut juga sebagai cacat las. Berikut adalah beberapa kategori dalam cacat las:

1. *Lack of penetration*
2. *Incomplete fusion*
3. *Porosity*
4. *Undercutting*
5. *Cracking*
6. *Excessive spatter*
7. *Over spatter*

Pada umumnya cacat las dalam suatu bidang tidak hanya satu kategori cacat melainkan bisa juga terdapat beberapa jenis cacat las. Setiap cacat harus mampu diidentifikasi agar tidak melebihi ketentuan yang telah ditetapkan, contoh acuan umum yang digunakan adalah the ASTM International, the Society of Automotive Engineers (SAE), the American Iron and Steel Institution (AISI), the American Welding Society (AWS) and the ASME International.

2.1.1 Incoplete Fusion (*Cold Lap*)

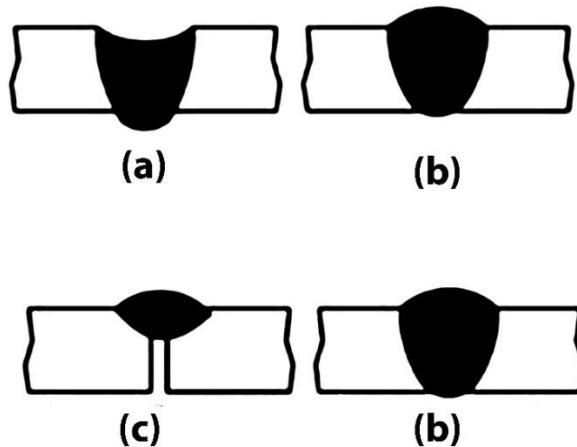
Incoplete Fusion adalah seberapa banyak titik antara bidang las (*base metal*) dengan material las (*weld metal*) yang tidak terhubung dengan sempurna (Jurandir Primo, 2012). Kesempurnaan dalam penggabungan antara dua bidang las atau lebih dengan material las sangat penting dalam menentukan kekuatan sambungan (*joint*). Contoh bentuk sambungan yang baik seperti yang ditunjukkan pada Gambar 2.1 point ke (a) dengan material las memenuhi seluruh bidang las.



Gambar 2.1 Bentuk sambungan las (*fusion*), (a) Good Fusion, (b) Poor Fussion , (c) Bridging Fusion, (d) Double Bevel Groove Welding With Incomplete Fussion, (e) Incomplete Fusion In A Grooved Butt Joint, (f) Incomplete Fusion One Log Of A Butt Joint

2.1.2 Lack of Penetration

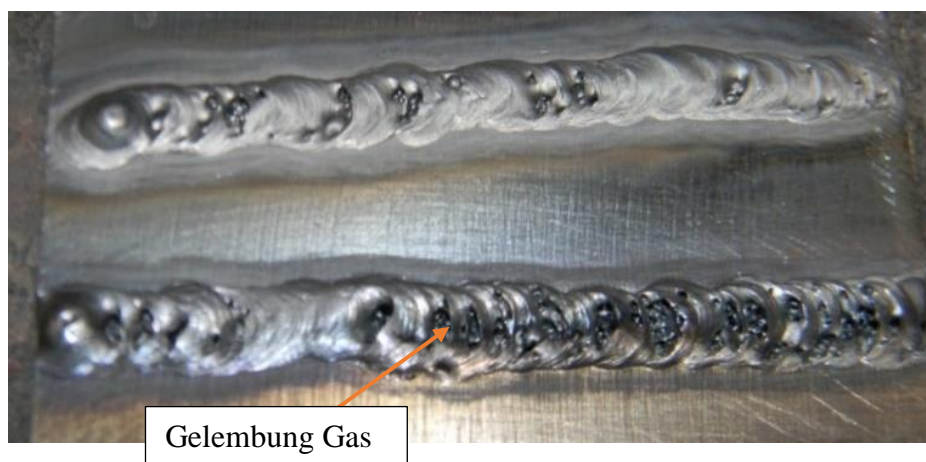
Cacat las jenis ini terjadi karena logam las gagal mencapai *root* (akar) dari sambungan dan gagal menyambungkan permukaan *root* secara menyeluruh. Hal ini disebabkan karena kesalahan dalam memilih ukuran elektroda, arus listrik yang terlalu kecil, dan rancangan sambungan yang kurang memadai. Kurang penetrasi sering dialami pada pengelasan posisi vertikal dan *overhead* (Jurandir Primo, 2012). Pada Gambar 2.2 a adalah contoh ketidak sempurnaan dalam proses penyambungan root dengan material las yang berlebihan sehingga melebihi bidanglas, sedangkan Gambar 2.2 c tidak memenuhi bidang las bagian bawah (root), sedangkan Gambar 2.2 b adalah penyambungan material las yang benar karena memenuhi seluruh bidang las.



Gambar 2.2 Jenis kegagalan penetrasi, (a) Excessive Penetration, (b) Good Penetration, (c) Lack of Penetration.

2.1.3 Porosity

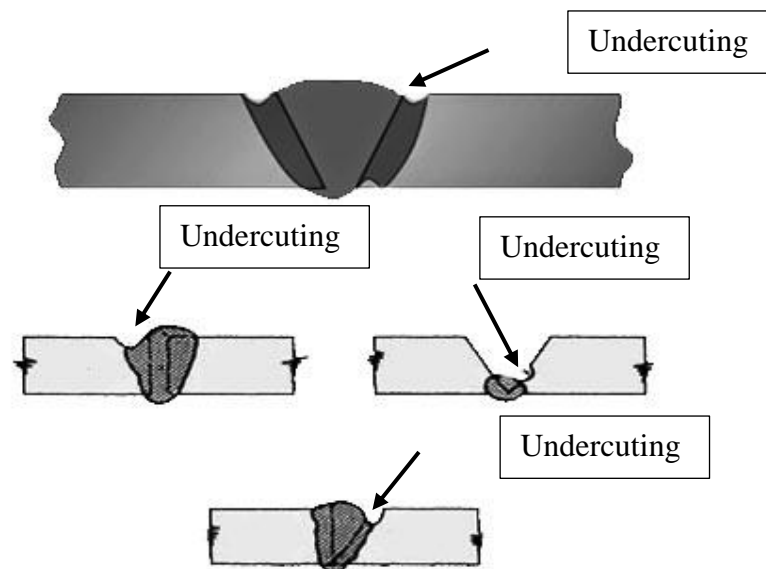
Porosity merupakan sekelompok gelembung gas yang terjebak di dalam material las. *Porosity* bisa terjadi karena proses pemadatan yang terlalu cepat. *Porosity* berupa rongga-rongga kecil berbentuk bola yang mengelompok pada lokasi-lokasi pengelasan. Terkadang terjadi rongga besar berbentuk bola tunggal atau tidak mengelompok. Rongga besar tersebut adalah *blow hole* (Jurandir Primo, 2012) seperti yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 *Welding porosity*

2.1.4 Undercutting

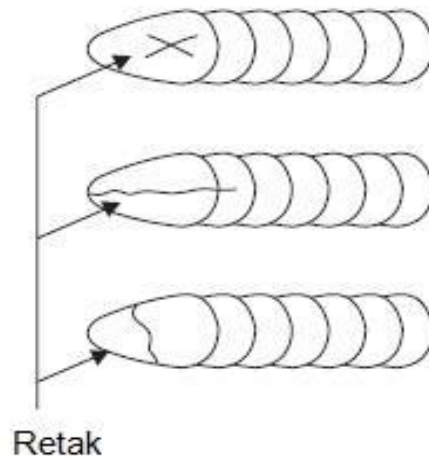
Undercutting merupakan istilah yang digunakan untuk menggambarkan sebuah alur (*groove*) benda kerja yang mencair dan terletak pada tepi atau kaki pengelasan (manik-manik las) dimana alur benda kerja yang mencair tersebut tidak terisi oleh cairan las. *Undercutting* disebabkan oleh arus yang berlebihan dalam proses pengelasan, kecepatan pengelasan yang terlalu tinggi, posisi elektrode saat pengelasan yang tidak tepat, dan ayunan elektrode saat pengelasan tidak teratur (Jurandir Primo, 2012). Cacat *undercutting* dapat dilihat pada Gambar 2.4 yang mana terdapat bidang las diarea alur terkikis saat proses pengelasan.



Gambar 2.4 *Welding undercut*

2.1.5 Craters

Retak (*craters*) merupakan putusnya benda kerja akibat tegangan. Retakan sering terjadi pada material las maupun bagian benda kerja yang dekat dengan bidang las. Retakan yang sering terjadi berupa retakan yang sangat sempit, walaupun tidak menutup kemungkinan terjadi retakan yang luas. Retakan dibagi dalam tiga jenis yakni: retakan panas, retakan dingin, dan *macrofissure*. Retakan yang terjadi pada material las dapat dilihat pada Gambar 2.5.



Gambar 2.5 Retakan pada pengelasan

2.1.6 *Over Spatter*

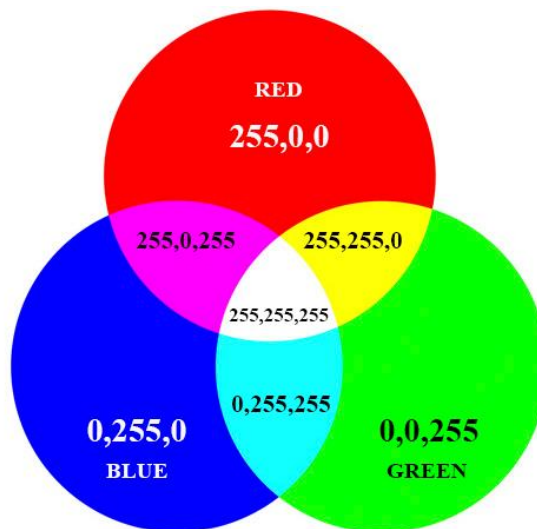
Over spatter adalah percikan las yang terlalu banyak memenuhi bidang las. *Over spatter* dapat disebabkan karena arus terlalu besar, busur las terlalu jauh, dan electrode meyerap banyak uap. *Over spatter* dapat menurunkan kerapian dalam pengelasan dan dapat menimbulkan terjadinya *porosity* (Jurandir Primo, 2012). *Over spatter* dapat dilihat pada Gambar 2.6.



Gambar 2.6 *Over spatter*

2.2 Model Warna RGB

Model warna RGB merupakan model warna yang paling umum digunakan pada pengolahan citra digital. Citra RGB disusun dari tiga buah kanal warna primer yaitu Merah (Red), Hijau (Green) dan Biru (Blue). Setiap kanal penyusun warna RGB berisikan intensitas warnanya dalam skala 8 bit atau range nilai antara 0 hingga 255. Pada tiap piksel elemen sebuah citra, berisikan perpaduan ketiga warna tersebut. Seperti pada Gambar 2.7 warna putih, perpaduan ketiganya adalah pada nilai maksimal (255,255,255) sedangkan untuk warna hitam, perpaduan warna ketiganya berada pada nilai minimal (0,0,0). Dari ketiga perpaduan warna ini didapatkan 16 juta warna variasi.



Gambar 2.7 Model Warna RGB

2.3 Model Warna Grayscale

Model warna keabuan (grayscale) merupakan warna dengan skala 8 bit atau range nilai antara 0 hingga 255. Warna keabuan bisa dikatakan sebagai warna dengan satu kanal warna, yaitu kanal warna 8 bit. Untuk mendapatkan citra dengan skala keabuan ini, dapat dilakukan pengubahan dari citra RGB ke grayscale. Hal yang paling umum dilakukan adalah dengan membagi tiga komposisi warna RGB tersebut. Masing-masing warna memiliki bobot 0.33, atau beberapa metode mengusulkan komposisi bobot tiap warnanya. Bergantung pada kebutuhan dan ketetapan rumusan standardnya. Gambar 2.8 adalah hasil konversi ruang warna

atau model RGB ke grayscale dengan menggunakan rata – rata nilai R, G, dan B setiap piksel.

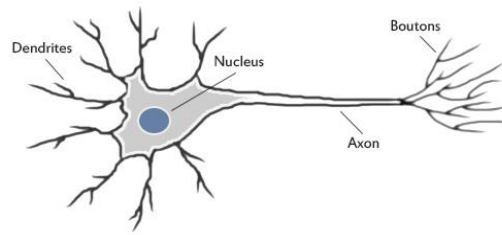


Gambar 2.8 Hasil Konversi Ruang Warna RGB ke Grayscale.

2.4 Neural Network (NN)

Neural network merupakan bagian dari sistem kecerdasan buatan yang digunakan untuk memproses informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya (Norvig, 1995). Pada Gambar 2.9 menunjukkan salah satu contoh syaraf secara biologis dimana setiap sel syaraf (Neuron) akan memiliki satu inti sel yang bertugas untuk melakukan pemrosesan informasi. Informasi yang datang akan diterima oleh dendrit.

Selain menerima informasi, Dendrit juga menyertai Axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini akan menjadi masukan bagi Neuron lain dimana antar Dendrit kedua sel tersebut dipertemukan dengan sinapsisnya. Informasi yang dikirimkan antar Neuron ini berupa rangsangan yang dilewatkan melalui Dendrit. Informasi yang datang dan diterima oleh Dendrit akan dijumlahkan dan dikirim melalui Axon lain. Informasi ini akan diterima oleh Neuron lain jika memenuhi batasan tertentu dikenal dengan nilai ambang (*threshold*) yang dikatakan teraktivasi.



Gambar 2.9 Syaraf secara biologi.

Pada umumnya jaringan syaraf tiruan memiliki tiga lapisan, yaitu *input layer*, *hidden layer*, dan *output layer*. Berikut penjelasan mengenai layer pada Neural Network.

1. Input Layer

Input layer berisi *Neuron-Neuron* yang masing-masing menyimpan sebuah nilai masukan yang tidak berubah pada fase latih dan hanya bisa berubah jika diberikan nilai masukan baru. *Neuron* pada lapisan ini tergantung pada banyaknya *input* dari suatu pola.

2. Hidden Layer

Lapisan ini tidak pernah muncul sehingga dinamakan *hidden layer*. Akan tetapi semua proses pada fase pelatihan dan fase pengenalan dijalankan di lapisan ini. Jumlah lapisan ini tergantung dari arsitektur yang akan dirancang, tetapi pada umumnya terdiri dari satu lapisan *hidden layer*.

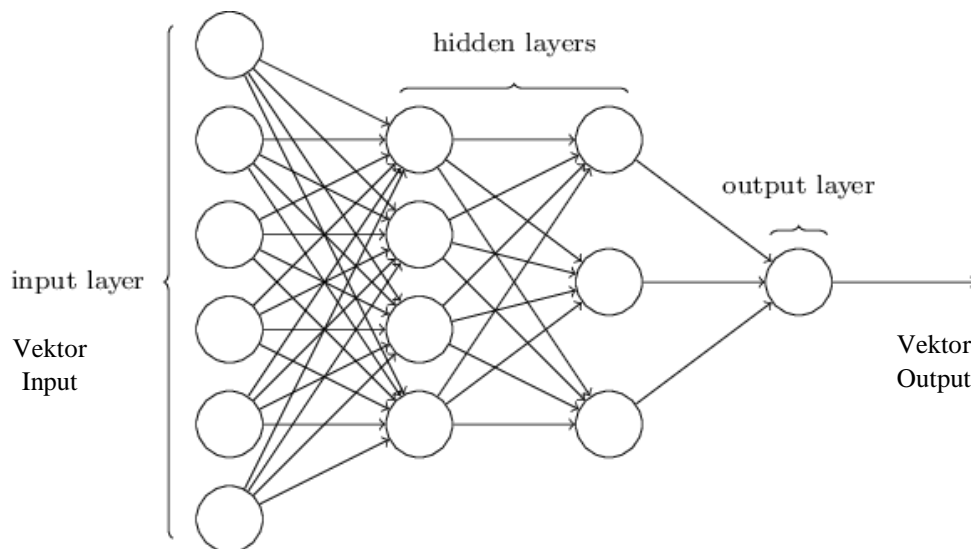
3. Output Layer

Output layer berfungsi untuk menampilkan hasil perhitungan sistem oleh fungsi aktivasi pada lapisan *hidden layer* berdasarkan *input* yang diterima.

Neural network ditentukan oleh tiga hal, yaitu pola hubungan antar-Neuron yang disebut jaringan. Metode untuk menentukan bobot penghubung yang disebut metode *training / learning / algoritma*, dan Fungsi aktivasi atau fungsi transfer. Salah satu arsitektur NN yang sangat populer adalah multi layer *feedforward networks*. Secara umum, jaringan seperti ini terdiri dari sejumlah unit Neuron sebagai lapisan input, satu atau lebih lapisan simpul-simpul Neuron komputasi

hidden (lapisan tersembunyi), dan sebuah lapisan simpul-simpul Neuron komputasi output. Sinyal input dipropagasikan ke arah depan (arah lapisan output), lapisan demi lapisan. Jenis jaringan ini adalah hasil generalisasi dari arsitektur perceptron satu lapisan, jadi biasa disebut sebagai Multi Layer Perceptron (MLP). Propagasi balik (ke arah lapisan input) terjadi setelah jaringan menghasilkan output yang mengandung error. Pada fase ini seluruh bobot synaptic (yang tidak memiliki aktivasi nol) dalam jaringan akan disesuaikan untuk mengoreksi/memperkecil error yang terjadi (error correction rule). Untuk pelatihan jaringan, pasangan fase propagasi ke depan dan balik dilakukan secara berulang untuk satu set data latihan, kemudian diulangi untuk sejumlah epoch (satu sesi lewatan untuk seluruh data latihan dalam sebuah proses pelatihan jaringan) sampai error yang terjadi mencapai batas kecil toleransi tertentu atau nol.

MLP terdiri dari beberapa unit pemroses (Neuron) seperti pada Gambar 2.10, yang terhubung dan mempunyai beberapa masukan serta memiliki satu atau beberapa output. Perceptron digunakan untuk menghitung jumlah nilai perkalian penimbang dan masukan dari parameter yang kemudian dibandingkan dengan nilai threshold, jika keluran lebih besar dari *threshold* maka keluaran adalah satu, sebaliknya adalah nol (Mauridhi Hery, 2006).



Gambar 2.10 Struktur MLP

Pernyataan ini merupakan hasil proses pelatihan yang dalam bentuk bahasanya adalah pernyataan iya atau tidak. Secara matematis dapat ditulis dengan persamaan berikut:

Jumlah perkalian penimbang dan paramater masukan adalah :

$$I = w_{ji} x_i \quad (2.1)$$

Dari persamaan 2.1, x_i adalah sinyal masukan sedangkan w_i adalah penimbang. Bila $I > T$ maka keluaran $O = 1$ sebaliknya $O=0$, dengan T adalah *threshold*. Pelatihan pada perceptron dilakukan dengan merubah nilai penimbangnya sehingga sesuai dengan kebutuhan, dilakukan dengan membandingkan keluaran dari JST dengan targetnya, proses tersebut dapat dituliskan pada persamaan 2.1:

$$w_{baru_{ji}} = w_{lama_{ji}} + \alpha(t_j - O_j)x_i \quad (2.2)$$

t_j adalah target dan α adalah bilangan konstan bernilai antara 0,1 sampai 0,9 yang disebut dengan laju pelatihan (*learning rate*). Proses diatas dijalankan pada setiap Neuron yang ada pada setiap layer sampai penimbang tersebut sesuai dengan yang diinginkan. Nilai awal penimbang adalah bilangan kecil yang diambil secara acak.

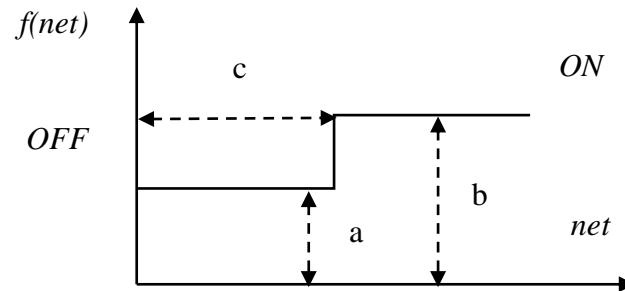
2.4.1 Fungsi Aktivasi

Melakukan aktivasi NN dimaksudkan untuk menjadikan setiap Neuron yang dipakai pada jaringan tersebut aktif. Banyak fungsi yang dipakai sebagai pengaktif seperti fungsi-fungsi trigonometri dan hiperboliknya, fungsi unit step, impuls, dan sigmoid. Tetapi pada umumnya yang digunakan adalah fungsi sigmoid (Mauridhi Hery, 2006). Pada umumnya fungsi pengaktif membangkitkan fungsi unipolar atau bipolar. Fungsi unit step seperti pada

Gambar 2.11 dan ramp seperti pada Gambar 2.12, fungsi ramp sering digunakan dalam NN yang menggunakan Neuron linier berfungsi sebagai elemen threshold dengan keluaran biner. Bila aktivasi sinapsis melebihi nilai thresholdnya maka fungsi step tersebut dapat ditambah dengan bias. Bias berlaku untuk unipolar dan bipolar. Fungsi sigmoid dapat digunakan untuk jenis unipolar seperti pada Gambar 2.13 dan maupun bipolar seperti pada Gambar 2.14 (Mauridhi Hery, 2006).

1. Fungsi Step

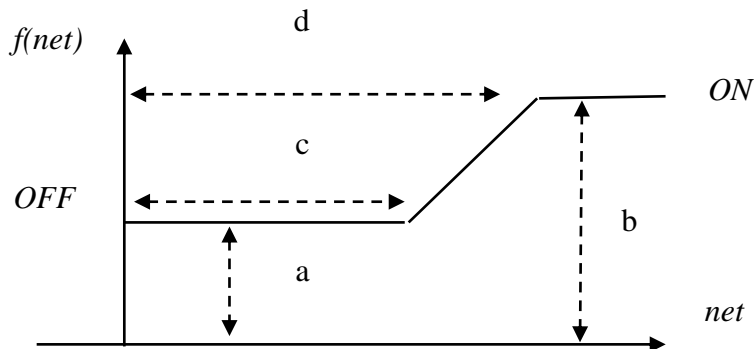
$$f(\text{net}) = \begin{cases} a & \text{bila } \text{net} < c \\ b & \text{bila } \text{net} \geq c \end{cases} \quad (2.3)$$



Gambar 2.11 Fungsi step

2. Fungsi Ramp

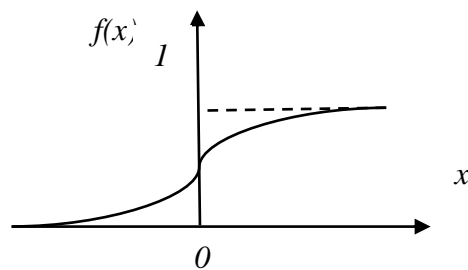
$$f(\text{net}) = \begin{cases} a & \text{bila } \text{net} \leq c \\ b & \text{bila } \text{net} \geq d \\ a + \frac{(\text{net} - c)(b - a)}{(d - c)} & \text{Selain kondisi tersebut di atas} \end{cases} \quad (2.4)$$



Gambar 2.12 Fungsi ramp

3. Fungsi Sigmoid Unipolar

$$f(x) = \left(\frac{1}{1 + e^{-f(x)}} \right) \quad (2.5)$$

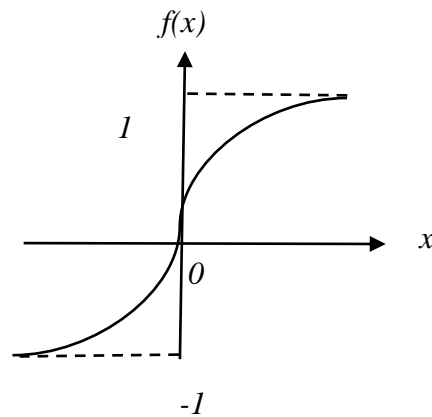


Gambar 2.13 Fungsi sigmoid unipolar

4. Fungsi Sigmoid Bipolar

$$f(x) = \left(\frac{1 - e^{-f(x)}}{1 + e^{-f(x)}} \right) \quad (2.6)$$

$$\begin{aligned} f(x) &= \left(\frac{e(x) - e(-x)}{e(x) + e(-x)} \right) \\ &= \frac{1 - e(-2x)}{1 + e(-2x)} \end{aligned}$$



Gambar 2.14 Fungsi sigmoid bipolar

2.4.2 Algoritma *Backpropagation*

Algoritma Pelatihan propagasi balik (*Backpropagation*) pertama kali dirumuskan oleh Werbos dan dipopulerkan oleh Rumelhart bersama McClelland, algoritma ini termasuk metoda pelatihan terbimbing (*supervised*) yang didesain untuk operasi pada NN *feedforward* lapis jamak (*multi-layer*) (Mauridhi Hery, 2006).

Secara garis besar alur dari metode *backpropagation* adalah ketika diberi masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk menuju ke unit-unit lapisan keluaran. Kemudian unit-unit lapisan keluaran memberikan tanggapan yang disebut dengan output NN. Saat keluar NN tidak sama dengan keluaran yang diharapkan maka keluaran akan disebarkan mundur (*backward*) pada lapis tersembunyi diteruskan ke unit pada lapisan masukan.

Algoritma pelatihan Backpropagation Neural Network (BPNN) terdiri atas dua tahapan yaitu, *feedforward* dan *back propagation* dari galatnya (Mauridhi Hery, 2006).

Algoritma Neural Network

Input:

Vektor Input $\leftarrow i$

Output:

Vektor Output

Inisialisasi jumlah Neuron input, hidden, dan Output.

Inisialisasi semua bobot dengan nilai random, antara -1 dan 1

Inisialisasi Epoc $\leftarrow n$

Inisialisasi maxError $\leftarrow m$

While (Epoc $\leq n$ || Nilai Error \geq maxError)

// propagasi maju setiap Neuron pada layer

For each Layer dalam jaringan

For each Neuron layer (Z)

1. Kalkulasi jumlah bobot V_{ij} dan bias V_{oj} setiap input X_i yang menuju hidden Neuron .

$$Z_{in_j} = V_{oj} + \sum_{i=1}^n X_i V_{ij} \quad (2.7)$$

2. Aktivasi setiap Neuron

$$Z_j = f(Z_{in_j}) \quad (2.8)$$

end

end

//Propagasikan balik nilai error output terhadap target

For each Neuron do output layer

Kalkulasi nilai error output (y_k) terhadap target (t_k)

$$\delta_k = (t_k - y_k) y_k (1 - y_k) \quad (2.9)$$

End

For all hidden layer

For each node pada layer

1. Kalkulasi perbaikan setiap jaringan node.

$$\Delta W_{jk} = \alpha \cdot \delta_k \cdot Z_j \quad (2.10)$$

2. Aktivasi setiap Neuron

$$Z_j = f(Z_{in_j}) \quad (2.11)$$

3. Update setiap bobot Neuron dalam jaringan

end

end

Kalkulasi total error (MSE)

end

2.5 Convolution Neural Network (CNN)

Secara umum konvolusi didefinisikan sebagai cara untuk mengkombinasikan dua buah deret angka yang menghasilkan deret angka yang ketiga. Secara matematis, konvolusi adalah integral yang mencerminkan jumlah lingkupan dari sebuah fungsi a yang digeser atas fungsi b sehingga menghasilkan fungsi c. Konvolusi dilambangkan dengan asterisk. Sehingga, $a \times b = c$ berarti fungsi a dikonvolusikan dengan fungsi b menghasilkan fungsi c.

Konvolusi diskrit banyak digunakan dalam pengolahan citra untuk image smoothing, edge detection dan efek-efek lainnya, konvolusi dilakukan berdasarkan jumlah bobot dari piksel-piksel tetangga dengan bobot ditentukan berdasarkan ukuran window berupa matriks. Window atau disebut juga sliding window bergerak sepanjang piksel yang ada pada citra berukuran kecil yang biasa disebut convolution mask atau convolution kernel. Orde matriks biasanya ganjil sehingga hasil konvolusi tepat berada ditengah-tengah, semakin besar ukuran window, beban komputasi akan semakin meningkat.

Konvolusi seringkali dilibatkan dalam operasi ketetanggaan piksel. Konvolusi pada citra sering disebut sebagai konvolusi dua-dimensi (konvolusi 2D). Konvolusi 2D didefinisikan sebagai proses untuk memperoleh suatu piksel didasarkan pada

nilai piksel itu sendiri dan tetangganya, dengan melibatkan suatu matriks yang disebut kernel yang merepresentasikan pembobotan. Terdapat empat fungsi konvolusi yang seriang dipergunakan dalam pengolahan citra yaitu Embossing, Blurring, menajamkan (sharpening), mendeteksi tepi (edge detection), berikut adalah penjelasannya. Gambar 2. 15 adalah hasil dari proses konvolusi dengan menggunakan kernel masing-masing fungsi.

1. Embossing

Embossing adalah kernel yang digunakan untuk mengubah citra menjadi gelap (shadow) dan terang (highlight), tergantung pada batas terang / gelap pada gambar aslinya. Area dengan kontras rendah diganti dengan latar belakang abu-abu atau hitam. Koefisien jendela konvolusi memiliki bobot tengah bernilai 0 & jumlah seluruh bobot = 0.

2. Blurring

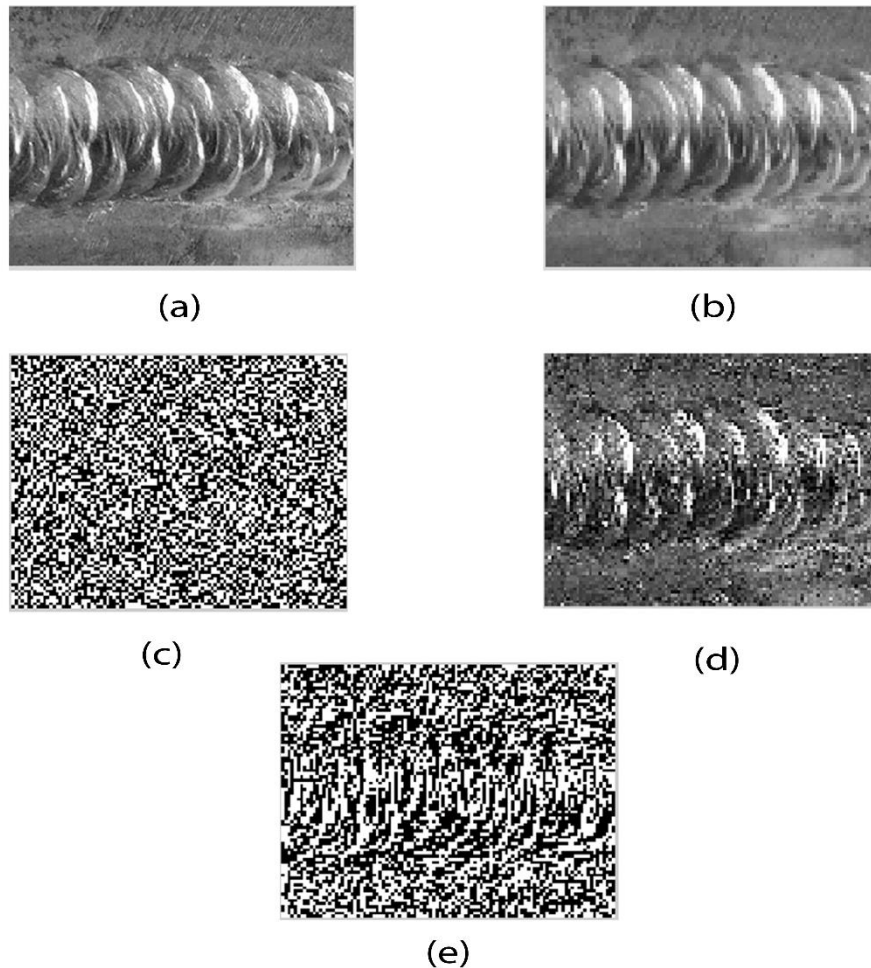
Blurring (Pengaburan) yaitu filter spasial low-pass yang melenyapkan detail halus dari suatu citra. Pengaburan dicapai melalui konvolusi dari seluruh koefisien mask bernilai sama. Blurring ini perataan nilai pixel-pixel tetangga, makin besar ukuran mask maka makin besar efek pengaburan

3. Sharpening

Sharpening (Penajaman) yaitu memperjelas detail suatu citra (menambah kontras) dengan penjumlahan atas citra tepi dengan citra aslinya maka bagian tepi objek akan terlihat berbeda dengan latarnya, sehingga citra terkesan lebih tajam.

4. Edge Detection

Deteksi tepi yaitu proses menentukan lokasi titik-titik yang merupakan tepi objek.



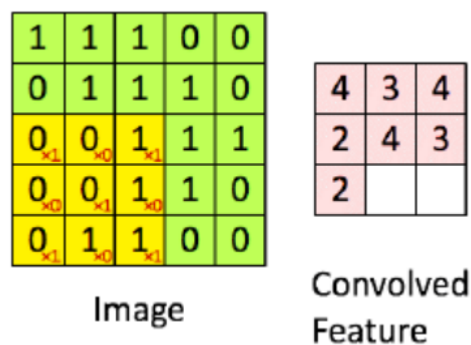
Gambar 2. 15 Citra Hasil Konvolusi, (a) Citra Asli, (b) Blurring, (c) Embossing, (d) Sharpening, (e) Edge.

Convolution Neural Network sendiri adalah perpaduan antara konvolusi citra untuk proses ekstraksi *feature*, dan Nural Network untuk klasifikasi. Berdasarkan aritektur LeNet5 (Stanford, 2016), terdapat 4 macam layer utama pada sebuah CNN yaitu convolution layer, relu layer, subsampling layer, dan fully connected layer. Fungsi layer awal sebagai metode ekstraksi fitur, maka jumlah layer dalam sebuah CNN tidak memiliki aturan universal dan berlaku berbedabeda tergantung dataset yang digunakan. Karena hal tersebut, jumlah layer pada jaringan serta jumlah Neuron pada masing-masing layer dianggap sebagai hyper parameter dan dioptimasi menggunakan pendekatan *searching*. Berikut ini adalah penjelasan mengenai masing-masing layer.

1. Convolution Layer

Convolution Layer melakukan operasi konvolusi pada output dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra disemua offset yang memungkinkan seperti yang ditunjukkan pada Gambar 2.16. kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi, kernel bergerak dari sudut kiri atas ke kanan bawah. Ukuran dimensi kernel pada penelitian ini memiliki ordo 3x3. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada konvolusi fitur.

Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.



Gambar 2.16 Operasi Konvolusi (I Wayan Suartika E. P, 2016).

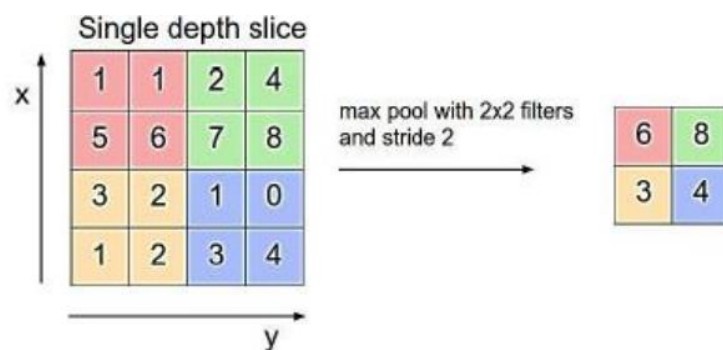
2. ReLu Layer

ReLu atau Rectified Linear Unit Layer, pada layer ini dapat diibaratkan seperti thresholding atau sama halnya seperti fungsi aktivasi pada jaringan syaraf tiruan. Dengan tujuan untuk menjaga hasil citra proses konvolusi berada pada

domain definit positif. Angka yang dihasilkan harus bernilai positif dikarenakan fungsi aktivasi pada jaringan syaraf tiruan propagasi balik pada penelitian ini menggunakan fungsi sigmoid. Sehingga setiap angka hasil proses konvolusi yang bernilai negatif akan terlebih dahulu melalui proses ReLu yang menjadikan nilai negatif bernilai sama dengan nol.

3. Subsampling layer

Subsampling adalah proses mereduksi ukuran sebuah data citra. Dalam pengolahan citra, subsampling juga bertujuan untuk meningkatkan invariansi posisi dari fitur. Dalam sebagian besar CNN, metode subsampling yang digunakan adalah max pooling. Max pooling membagi output dari convolution layer menjadi beberapa grid kecil lalu mengambil nilai maksimal dari setiap grid untuk menyusun matriks citra yang telah direduksi seperti yang ditunjukkan pada Gambar 2.17. Grid yang berwarna merah, hijau, kuning dan biru merupakan kelompok grid yang akan dipilih nilai maksimumnya. Sehingga hasil dari proses tersebut dapat dilihat pada kumpulan grid disebelah kanannya. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran).



Gambar 2.17 Operasi Max Pooling (I Wayan Suartika E. P, 2016).

Menurut Springenberg (J. T. Springenberg, 2015), penggunaan *pooling layer* pada CNN hanya bertujuan untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan sebuah *convolution layer* dengan *stride* yang sama dengan pooling layer yang bersangkutan.

4. Fully Connected Layer

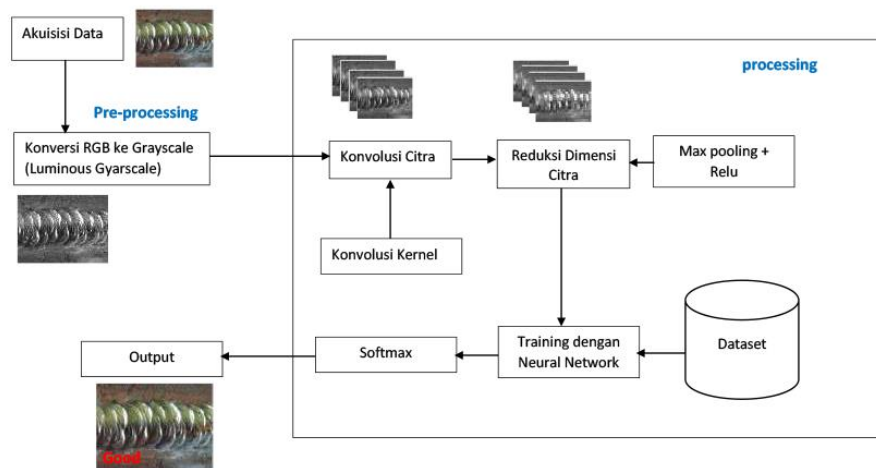
Layer tersebut adalah layer yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap Neuron pada convolution layer perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah fully connected layer. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, fully connected layer hanya dapat diimplementasikan di akhir jaringan. Convolution layer dengan ukuran kernel 1×1 melakukan fungsi yang sama dengan sebuah *fully connected layer* namun dengan tetap mempertahankan karakter spasial dari data.

Halaman ini sengaja dikosongkan

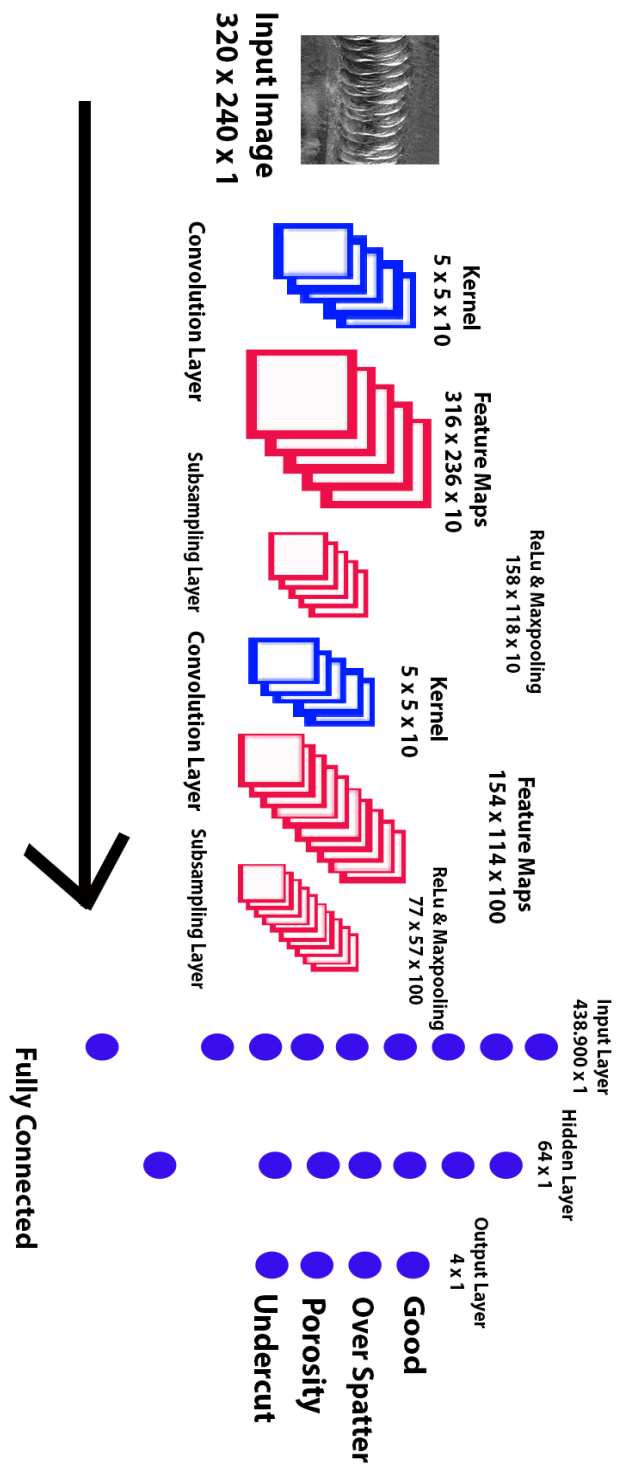
BAB 3

METODOLOGI PENELITIAN

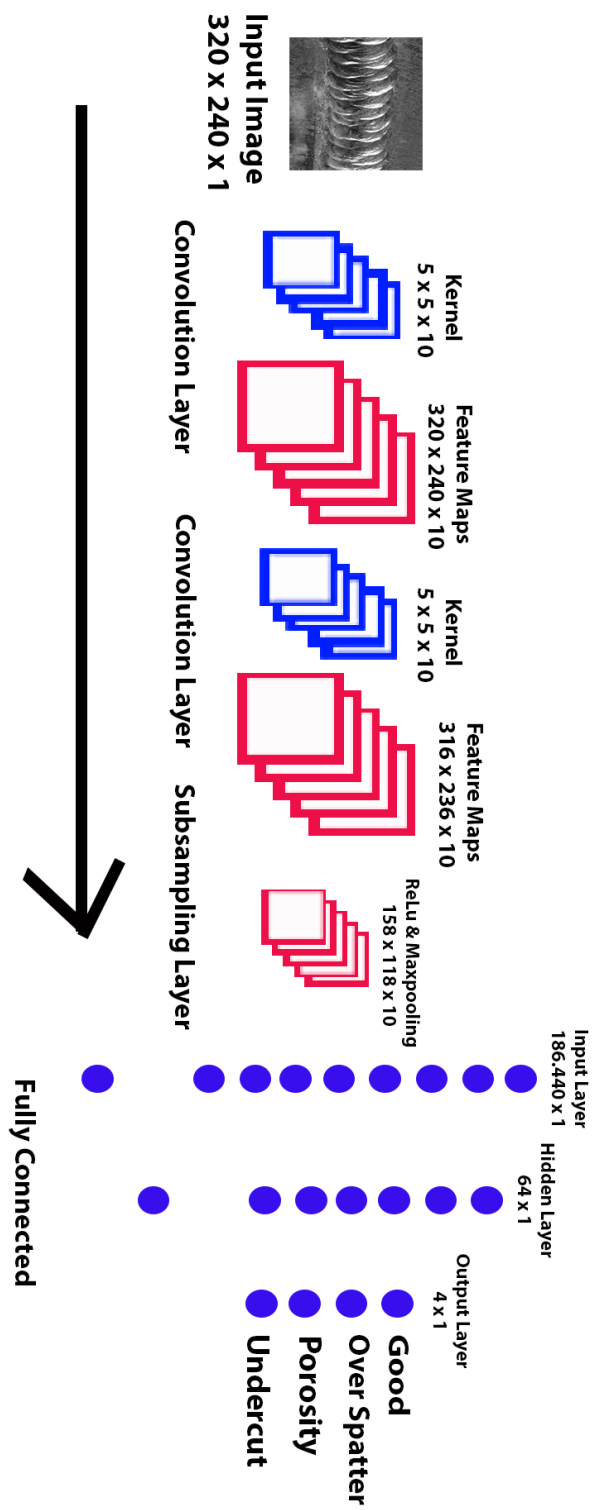
Penelitian ini difokuskan untuk mendapatkan klasifikasi hasil pengelasan dengan input data berupa *image sequence* atau video. Adapun proses CNN yang digunakan pada image squence dapat dilakukan melalui 2 aliran proses seperti pada yaitu Spatial Stream Convolution Net dan Temporal Stream Convolution Net, Spatial stream convolution adalah mengambil semua bagian citra untuk dilakukan proses konvolusi, sedangkan Temporal Stream Convolution hanya mengambil bagian citra tertentu (Karen Simonyan, 2014). Pada penelitian ini aliran proses yang digunakan adalah Spatial Stream Convolution Net dengan Setiap tahap konvolusi citra dengan masing-masing tahapan melalui proses normailisasi (Rectified Linear Unit/Relu) serta proses subsampling menggunakan maxpooling, tahapan selanjutnya adalah proses training dengan menggunakan Neural Network tahapan ini citra dari *image sequence* akan dibandingkan dengan dataset yang sudah dikodekan dan dilabel. Output data berupa nilai error dari masing-masing kategori data pada dataset (softmax) metode yang digunakan dalam penelitian seperti pada Gambar 3.1. Pada penelitian ini terdapat 2 arsitektur CNN yang digunakan yang kemudian akan dibandingkan tingkat keberhasilannya, kedua arsitektur CNN yang digunakan dapat dilihat pada Gambar 3.2 dan Gambar 3.3 .



Gambar 3.1 Metode Penelitian.



Gambar 3.2 Desain Arsitektur CNN 1 (Stanford, 2016).



Gambar 3.3 Desain Arsitektur CNN 2

3.1 Metode Usulan

Pada penelitian ini terdapat 2 arsitektur CNN yang digunakan yang kemudian akan dibandingkan tingkat keberhasilannya, untuk arsitektur yang pertama dapat dilihat pada Gambar 3.2. arsitektur CNN pertama yang digunakan pada penelitian ini adalah single hidden layer dengan jumlah Neuron sebanyak 64 hidden Neuron serta menggunakan fungsi aktivasi sigmoid di setiap layernya. Arsitektur CNN yang pertama meliputi proses konvolusi, subsampling + Relu, dan klasifikasi menggunakan *Backpropagation* NN. Untuk arsitektur yang pertama proses konvolusi dan subsampling masing-masing melewati dua kali perulangan.

Gambar 3.3 menjelaskan struktur detail arsitektur CNN yang digunakan pada, dimulai dari citra yang digunakan adalah citra grayscale berukuran 320 x 240. Citra ini diproses melalui kernel konvolusi yaitu kernel Gaussian berukuran 5 x 5 dengan jumlah fitur konvolusi sebanyak 10 sehingga dihasilkan 10 citra baru berukuran lebih kecil yaitu 316 x 236 piksel. Proses selanjutnya adalah subsampling yang merupakan proses linearisasi atau ReLu untuk mengubah nilai domain negatif menjadi domain positif untuk kemudian disederhanakan menggunakan operasi maxpool berukuran 2 x 2 sehingga citra hasil subsampling berukuran 158 x 118 sebanyak 10 citra.

Proses konvolusi diulang untuk kedua kalinya dengan ukuran kernel Gaussian yang sama yaitu 5 x 5 sehingga didapatkan 100 citra baru berukuran 154 x 114 piksel, dan setelah melalui proses subsampling 2 x 2 didapatkan hasil akhir berupa 100 citra baru berukuran 77 x 57 x 100. Proses konvolusi dan subsampling tidak dapat diulang karena dimensi matriks yang berupa bilangan ganjil sehingga akan terhenti ketika operasi subsampling 2 x 2.

Dimensi 77 x 57 x 100 atau setara dengan 438.900 vektor input dengan jumlah input sebanyak 100 citra. Setiap vektor input akan di kalkulasi dengan bobot yang dihasilkan secara acak menuju 64 Neuron pada lapisan tersembunyi dan menghasilkan nilai Z_j yang merupakan keluaran Neuron tersembunyi yang sudah melalui proses aktivasi sigmoid. Kalkulasi keseluruhan nilai Z_j pada tiap Neuron tersembunyi menghasilkan nilai y_k setelah hasil penjumlahan Z_j melalui proses

aktivasi sigmoid. Jika nilai y_k belum memenuhi target yang diharapkan maka proses propagasi balik berlangsung dengan parameter laju pembelajaran α sebagai konstanta perbaikan bobot setiap layer hingga nilai y_k memenuhi atau mendekati nilai target (t).

Sedangkan untuk arsitektur CNN 2 sedikit berbeda dalam proses ekstraksi fiturnya, yaitu proses konvolusi yang diulang sebanyak 2 kali sedangkan subsampling layernya hanya diulang 1 kali. Sama seperti dengan arsitektur CNN 1 Citra grayscale diproses melalui kernel konvolusi yaitu kernel Gaussian berukuran 5×5 dengan jumlah fitur konvolusi sebanyak 10 sehingga dihasilkan 10 citra baru berukuran sama tanpa mengurangi pixel yaitu 320×240 piksel, kemudian proses konvolusi diulang untuk yang kedua kalinya dengan kernel 5×5 pada konvolusi ini *feature map* tidak bertambah karena masing-masing *feature map* hanya dilakukan konvolusi sebanyak satu kali sehingga untuk *feature map* yang kedua berukuran $316 \times 236 \times 10$. Proses selanjutnya adalah subsampling dan ReLu dengan reduksi setengah kali ukuran pixel image sebelumnya oleh karena itu pada layer ini *feature map* akan berukuran $158 \times 118 \times 10$. Setelah proses ekstraksi *feature* maka dilakukan flatting yaitu menjadikan matrik 2 D ($m \times n$) menjadi 1 D ($m \times 1$), proses flatting ini menghasilkan vektor input sebanyak 186.440 yang kemudian akan melalui proses training dengan 4 vektor output yaitu *good*, *porosity*, *Undercut*, dan *over spatter*.

3.2 Tahapan Pelatihan Dataset

Untuk mendapatkan hasil klasifikasi yang tinggi dibutuhkan dataset yang sangat banyak untuk masing-masing vektor input, Proses training terdiri atas dua proses yaitu *feedforward* dan *backpropagation*. Contoh citra yang akan dilatih seperti pada Gambar 3.4, dengan alur proses pelatihan seperti pada Gambar 3.5. Output *feedforward* berupa bobot yang akan digunakan untuk proses klasifikasi menggunakan Neural Network.

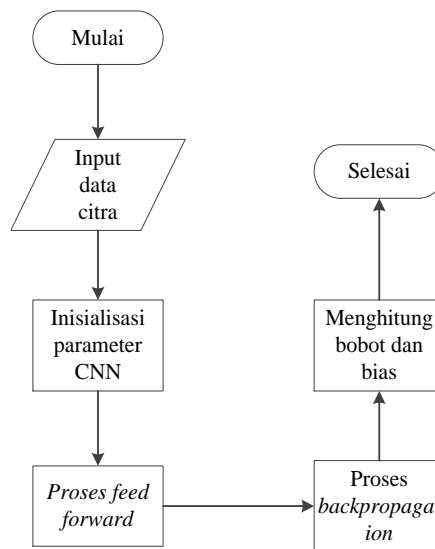
Dapat dilihat pada Gambar 3.2 dan Gambar 3.3 dimana semua dataset citra akan melalui proses konvolusi dengan menggunakan kernel ordo 5×5 . Total data

secara keseluruhan adalah 120 data dengan rincian data citra untuk kelas *good* sebanyak 30 data, data citra untuk kelas *over spatter* sebanyak 30 data, data citra untuk kelas *porosity* sebanyak 30 data dan terakhir untuk kelas *undercut* sejumlah 30 data. Hasil konvolusi kemudian akan direduksi dengan menggunakan max pooling dan memperbanyak Neuronnya. Dari Gambar 3.3 diketahui bahwa dataset citra yang digunakan mempunyai ruang warna R,G, dan B akan diproses menjadi ruang warna grayscale dengan resolusi 320 x 240 yang kemudian akan melalui proses konvolusi dengan menggunakan Gaussian kernel. Konvolusi bertujuan untuk ekstraksi fitur dari citra, semakin banyak kernel yang digunakan maka akan memperkaya *feature map* dan meningkatkan akurasi, output konvolusi layer yang pertama berdimensi 316 x 236 x 10 untuk CNN 1 dan 320 x 240 x 10 (P x L x D) untuk CNN 2. Proses selanjutnya adalah subsampling layer pada layer ini terdiri atas 2 proses yaitu Relu dan Maxpooling, Relu ini sama halnya seperti fungsi aktivasi pada jaringan syaraf tiruan, dengan tujuan untuk menjaga hasil citra proses konvolusi berada pada domain definit positif. Sedangkan Maxpooling adalah mencari nilai maksimal diarea tertentu, Maxpooling bertujuan untuk mereduksi ukuran citra, pada tahap ini citra akan direduksi menjadi 77 x 57 x 100 untuk CNN 1 dan 158 x 118 x 10 untuk CNN2.

Tahap selanjutnya proses flatting, untuk proses training jumlah Neuron vektor input adalah 438.900 untuk arsitektur CNN 1 dan 186.440 untuk arsitektur CNN 2, sedangkan 4 untuk vektor output untuk keduanya.



Gambar 3.4 Contoh citra hasil pengelasan baik (a) dan cacat las (b).



Gambar 3.5 Alur proses pelatihan.

Seperti yang ditunjukkan pada Gambar 3.5 bahwa proses pelatihan terdiri atas 3 tahapan yaitu:

1. Proses *feedforward*

Proses ini akan menghasilkan beberapa lapisan untuk mengklasifikasi data citra yang mana menggunakan bobot dan bias yang telah diperbarui dari proses *backpropagation*. Tahap ini juga akan digunakan kembali saat proses pengujian.

2. Proses *Backpropagation*

Proses *backpropagation* merupakan tahap kedua dari proses *training*. Pada tahap ini hasil proses dari *feedforward* dicari kesalahannya dari lapisan output sampai lapisan pertama. Untuk menandai bahwa data tersebut telah dicari dan diperoleh bobot dan bias yang baru.

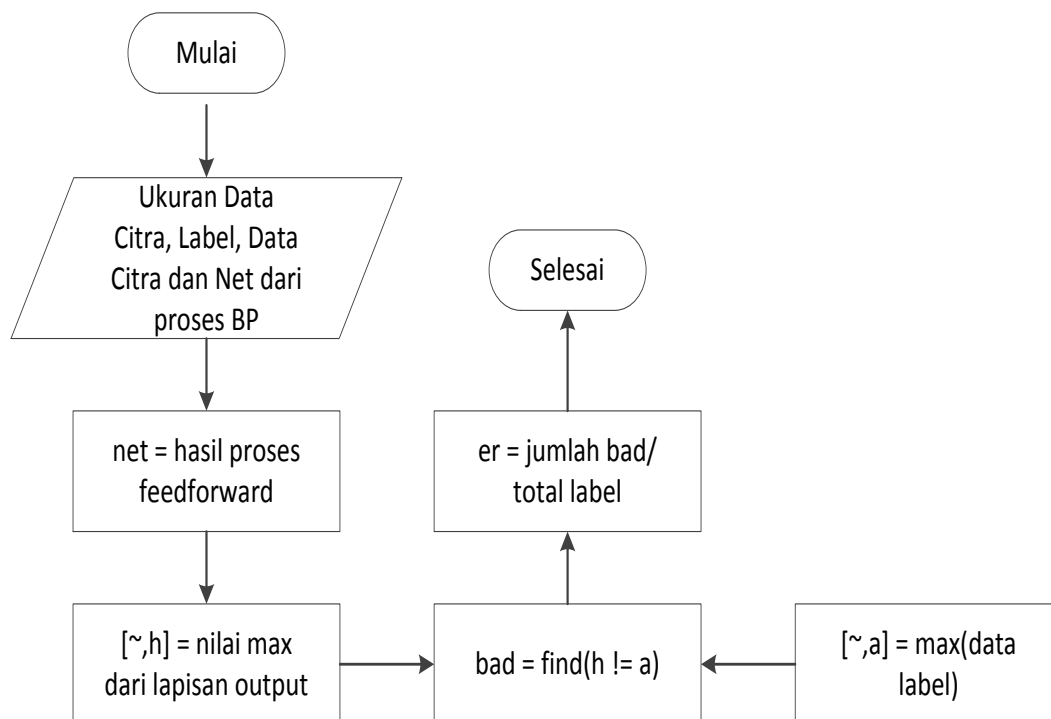
3. Perhitungan Gradien

Pada proses gradien untuk jaringan konvolusi merupakan proses untuk memperoleh nilai bobot dan bias yang baru yang akan diperlukan saat *training*.

3.3 Tahapan Pengujian Data

Proses pengujian merupakan proses klasifikasi menggunakan bobot dan bias dari hasil proses *training*. Proses ini tidak jauh berbeda dengan proses *training* yang membedakannya tidak terdapat proses *backpropagation* setelah proses *feedforward*. Sehingga hasil akhir dari proses ini menghasilkan akurasi dari klasifikasi yang dilakukan, data yang gagal diklasifikasi, nomor citra yang gagal diklasifikasi, dan bentuk network yang terbentuk dari proses *feedforward*. Dengan bobot dan bias yang baru proses *feedforward* diterapkan yang kemudian menghasilkan lapisan output.

Lapisan output sudah *fully connected* dengan label yang disediakan. Hasil *fully connected* tersebut diperoleh nilai error pada masing-masing vektor output, pada penelitian ini terdiri dari 4 vektor output yaitu *good*, *over spatter*, *porosity*, dan *undercut*. Dari penjelasan di atas bentuk alur proses pengujian berbentuk seperti pada Gambar 3.6.



Gambar 3.6 Tahapan Pengujian.

3.4 Konversi Data RGB to Gray

Format citra digital yang dipakai adalah citra skala keabuan. Format citra ini disebut skala keabuan karena pada umumnya warna yang dipakai warna hitam sebagai warna minimal (0) dan warna putih (255) sebagai warna maksimalnya, sehingga warna antaranya adalah abu-abu. Jadi dapat disimpulkan bahwa grayscale adalah warna perpaduan antara hitam dengan putih, yang mana grayscale nantinya juga memiliki tingkat derajat keabuan tersendiri. Terdapat tiga macam metode algoritma untuk mengubah nilai R G B menjadi grayscale, diantaranya:

1. *Lightness*

Algoritma ini adalah mencari nilai tertinggi dan terendah dari nilai R G B, kemudian nilai tertinggi dan terendah tersebut dijumlahkan kemudian dikalikan dengan 0.5 sehingga secara matematis dapat dirumuskan:

$$\text{Grayscale} = (\max(R, G, B)) + (\min(R, G, B)) * 0.5 \quad (3.1)$$

2. *Average*

Algoritmanya adalah dengan menjumlahkan seluruh nilai R G B, kemudian dibagi 3, sehingga diperoleh nilai rata-rata dari R G B, nilai rata-rata itulah yang dapat dikatakan sebagai grayscale. Rumus matematisnya adalah:

$$\text{Grayscale} = (R + G + B)/3 \quad (3.2)$$

3. *Luminosity*

Algoritmanya adalah dengan mengalikan setiap nilai R G B dengan konstanta tertentu yang sudah ditetapkan nilainya, kemudian hasil perkalian seluruh nilai R G B dijumlahkan satu sama lain. Rumus matematisnya adalah:

$$\text{Grayscale} = (0.21 \times R) + (0.72 \times G) + (0.07 \times B) \quad (3.3)$$

Dari ketiga macam algoritma diatas, masing-masing metode walaupun sama-sama menghasilkan warna grayscale, namun memiliki tingkat derajat keabuan yang berbeda, sehingga setiap metode menghasilkan warna keabuan yang tidak sama persis satu sama lain.

3.5 Proses Konvolusi

Sebelum proses konvolusi berlangsung citra atau gambar yang didapat berupa empat kondisi las yaitu *good*, *over spatter*, *undercut* dan *porosity* terlebih dahulu dikonversi kedalam bentuk grayscale. Dengan menjumlahkan nilai pixel setiap layer R, G dan B dan kemudian dibagi dengan nilai 3. Citra asli berukuran 240x320x3 (karena memiliki 3 layer RGB) setelah dikonversi menjadi 240x320x1.

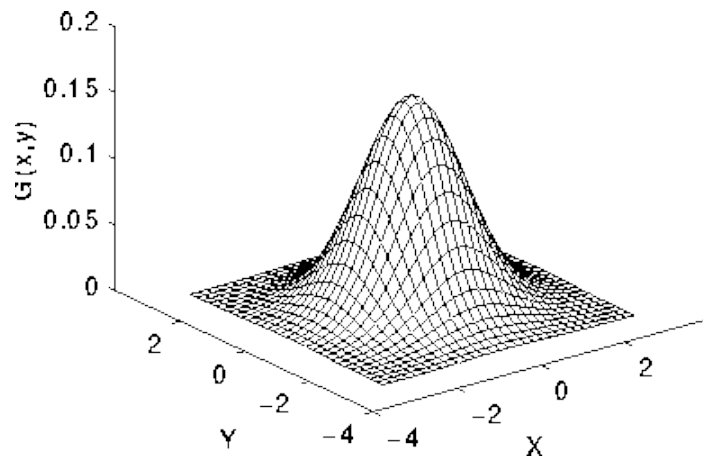
Citra grayscale tersebut kemudian akan melalui sebuah proses yang dinamakan konvolusi, proses konvolusi dilakukan dengan konsep sliding window berturut – turut dari kiri ke kanan, Gambar 3.3 contoh proses pengolahan dari citra grayscale yang semula berukuran 320 x 240 menjadi 316 x 236 pixel. Proses konvolusi merupakan ekstraksi pola citra asli ke dalam bentuk citra baru melalui satu atau lebih kernel. Pada saat proses capture (pengambilan gambar), ada beberapa gangguan yang mungkin terjadi, seperti kamera tidak fokus atau munculnya bintik-bintik yang bisa jadi disebabkan oleh proses capture yang tidak sempurna. Setiap gangguan pada citra dinamakan dengan noise. Kernel yang digunakan pada penelitian ini adalah kernel Gaussian atau kernel yang memiliki ciri distribusi normal secara statistik. Kernel Gaussian dipilih karena kemampuannya untuk menyamarkan citra atau ‘blur’ serta meminimalkan terjadinya noise pada citra baru yang dihasilkan. berikut adalah persamaan kernel Gaussian.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.4)$$

σ adalah lebar dari kernel gaussian, secara statistik simbol ini biasa disebut dengan standar deviasi atau simpangan baku dan hasil kuadratnya adalah varians. σ pada

penelitian ini digunakan sebagai penskalaan, semakin besar σ maka amplitude titik puncak kurva gaussian semakin berkurang.

Pada umumnya pola Gaussian memiliki dimensi bilangan ganjil seperti 3 x 3, 5 x 5, dan seterusnya. Pada kernel Gaussian bagian titik tengah matriks atau titik puncaknya memiliki nilai bilangan atau amplitude yang paling tinggi sedangkan daerah yang mendekati titik pusat akan mempunyai nilai yang mendekati nilai tertingginya, sedangkan apabila kordinat x dan y nya semakin menjauhi titik pusat, maka nilai akan semakin rendah, hal ini dapat dilihat seperti pada Gambar 3.7 . Pola distribusi normal seperti ini dapat menghasilkan citra baru tanpa menghilangkan pola asli dari gambar sebelumnya sekaligus meminimalkan noise. Untuk konfigurasi persamaan Gaussian terhadap posisi x dan y dapat dilihat pada Gambar 3.8.



Gambar 3.7 Grafik Perubahan Nilai Gaussian 2D Terhadap Koordinat X dan Y

$\frac{1}{2\pi\sigma^2} e^{-\frac{1^2+1^2}{2\sigma^2}}$	$\frac{1}{2\pi\sigma^2} e^{-\frac{1^2+0^2}{2\sigma^2}}$	$\frac{1}{2\pi\sigma^2} e^{-\frac{1^2+1^2}{2\sigma^2}}$
$\frac{1}{2\pi\sigma^2} e^{-\frac{0^2+1^2}{2\sigma^2}}$	$\frac{1}{2\pi\sigma^2} e^{-\frac{0^2+0^2}{2\sigma^2}}$	$\frac{1}{2\pi\sigma^2} e^{-\frac{0^2+1^2}{2\sigma^2}}$
$\frac{1}{2\pi\sigma^2} e^{-\frac{1^2+1^2}{2\sigma^2}}$	$\frac{1}{2\pi\sigma^2} e^{-\frac{1^2+1^2}{2\sigma^2}}$	$\frac{1}{2\pi\sigma^2} e^{-\frac{1^2+1^2}{2\sigma^2}}$

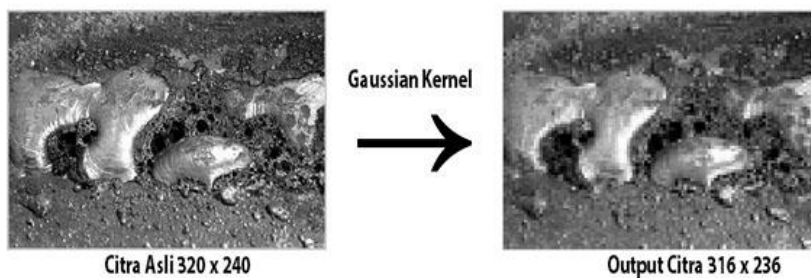
Gambar 3.8 Konfigurasi Persamaan Gaussian Dengan Koordinat X dan Y

Noise pada citra tidak hanya terjadi karena ketidak-sempurnaan dalam proses capture, tetapi bisa juga disebabkan oleh kotoran-kotoran yang terjadi pada citra. Beberapa contoh noise diantaranya noise Gaussian yang merupakan model noise yang mengikuti distribusi normal dengan rata-rata nol dan standard deviasi 1. Efek dari gaussian noise ini, pada gambar muncul titik-titik berwarna yang jumlahnya sama dengan persentase noise. Noise speckle merupakan model noise yang memberikan warna hitam pada titik yang terkena noise. Serta noise salt & pepper akan memberikan warna putih pada titik yang terkena noise. Pada beberapa pengolahan citra, terkadang untuk menguji suatu algoritma untuk dapat mereduksi noise, maka noise dihasilkan dari proses pembangkitan noise. Untuk membangkitkan noise digunakan suatu bilangan acak sebagai pengganti noise yang dihasilkan. Proses konvolusi berlangsung dengan menjumlahkan hasil perkalian tiap komponen citra grayscale dengan kernel Gaussian, karena matriks kernel Gaussian sehingga matriks citra grayscale yang diambil berukuran sama dengan matriks kernel.

Gambar 3.9 adalah citra baru hasil konvolusi dengan menggunakan kernel 5 x 5 dan standar deviasi σ adalah 2:

$$I' = H \otimes I \quad (3.5)$$

$$I'(x,y) = \sum_{i=-n}^n \sum_{j=-n}^m H(i,j) I(x+i,y+j)$$



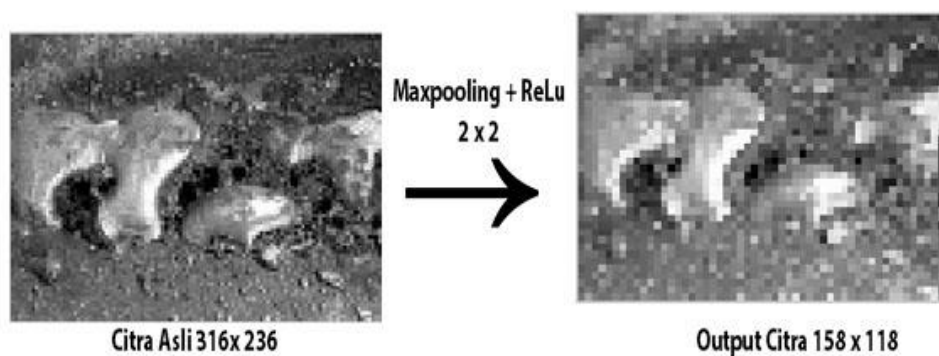
Gambar 3.9 Citra Baru Hasil Konvolusi

Matriks kernel Gaussian penelitian ini berukuran 5×5 dengan jumlah kernel Gaussian sebanyak 10 kernel yang dihasilkan secara acak. Citra hasil keseluruhan dari hasil perkalian dengan kernel Gaussian disebut sebagai *feature maps*.

3.6 Proses ReLu dan Maxpooling

Proses berikutnya adalah ReLu atau Rectified Linear Unit, proses ini sama halnya seperti fungsi aktivasi pada jaringan syaraf tiruan. Dengan tujuan untuk menjaga hasil citra proses konvolusi berada pada domain definit positif. Angka yang dihasilkan harus bernilai positif dikarenakan fungsi aktivasi pada jaringan syaraf tiruan propagasi balik pada penelitian ini menggunakan fungsi sigmoid. Sehingga setiap angka hasil proses konvolusi yang bernilai negatif akan terlebih dahulu melalui proses ReLu yang menjadikan nilai negatif bernilai sama dengan nol.

Ukuran citra yang masih besar tentunya menjadi kendala pada saat proses training menggunakan Neural Network oleh sebab itu ukuran citra hasil proses linearisasi (ReLu) dikecilkan melalui proses maxpooling. Maxpooling merupakan proses penyederhanaan matriks yang dilakukan berulang setiap $n \times m$ matriks. Proses tersebut diulangi secara berturut – turut hingga didapatkan citra baru. Dengan catatan jika citra hasil konvolusi merupakan kelipatan ganjil tidak dapat diproses pada tahap maxpooling dengan matriks kelipatan genap, hasil operasi max pooling dan Relu seperti pada Gambar 3.10.



Gambar 3.10 Citra Baru Hasil Proses ReLu dan Maxpooling

Proses Relu dan Maxpooling disebut subsampling. Proses konvolusi dan subsampling dilakukan berulang kali hingga citra yang dihasilkan tidak dapat di-*resize*, sebagai contoh citra 13 x 13 yang melalui proses konvolusi dengan matriks kernel Gaussian berukuran 5 x 5 menjadi citra baru 9 x 9, citra baru hasil konvolusi tersebut tidak dapat melalui proses subsampling dikarenakan matriks maxpooling berukuran 2 x 2.

3.7 Fully Connected

Citra baru hasil subsampling terakhir akan dijadikan masukan pada proses training. Sebagai contoh misalkan hasil subsampling terakhir pada Gambar 2.17 Operasi Max Pooling menghasilkan citra berukuran 158 x 118, sehingga jumlah vektor input untuk NN adalah 18.644. Proses dari matriks subsampling hingga menjadi komponen bilangan individu sebagai masukan neural network dinamakan flattening.

Proses training dengan menggunakan Backpropagation Neural Network (BPNN). *Backpropagation* merupakan salah satu model jaringan yang populer pada jaringan syaraf tiruan. Model jaringan ini banyak digunakan untuk diaplikasikan pada penyelesaian suatu masalah berkaitan dengan identifikasi, prediksi, pengenalan pola dan sebagainya. Pada latihan yang berulang-ulang, algoritma ini akan menghasilkan unjuk kerja yang lebih baik. Hal ini berarti bahwa “bobot interkoneksi” NN semakin mendekati bobot yang seharusnya. Kelebihan lain yang dimiliki NN ini adalah kemampuannya untuk belajar (bersifat adaptif) dan kebal terhadap adanya kesalahan (Fault Tolerance) dengan kelebihan tersebut NN dapat mewujudkan sistem yang tahan akan kerusakan (robust) dan konsisten bekerja dengan baik.

Secara garis besar proses pelatihan pada jaringan saraf tiruan dikenal beberapa tipe pelatihan, yaitu Supervised Training, Unsupervised Training, Fixed-Weight Nets. Metode pelatihan *backpropagation* atau dikenal dengan Generalize Delta Rule (GDR) ini merupakan supervised training dimana untuk tiap pola input terdapat pasangan target output untuk masing-masing pola input. Sebenarnya adalah metode gradient descent untuk meminimalkan total square error pada

keluaran hasil perhitungan jaringan. Ide dasarnya dapat dideskripsikan dengan pola hubungan yang sederhana yaitu: jika output memberikan hasil yang tidak sesuai dengan target yang tidak diinginkan, maka bobot akan dikoreksi agar errornya dapat diperkecil dan selanjutnya respon jaringan diharapkan akan lebih mendekati harga yang sesuai. Pada umumnya tujuan jaringan syaraf tiruan melakukan proses pelatihan adalah untuk mendapatkan *balancing* antara kemampuan jaringan untuk menanggapi secara benar pola-pola input pada saat pelatihan (dapat dikatakan kemampuan mengingat) dan kemampuan untuk memberikan penilaian yang layak dari suatu pola masukkan lain yang serupa. Sehingga dari proses pelatihan tersebut akan dibentuk suatu harga pembobot yang akan digunakan sebagai faktor penggali dari pola masukkan yang lain.

Pada metode ini, terdapat tiga tahapan dalam proses pelatihan, yaitu: proses umpan maju dari pola input pelatihan, perhitungan dan propagasi balik dari error yang terjadi dan penyesuaian nilai bobot. Pada tahap pelatihan ini merupakan langkah bagaimana suatu jaringan syaraf itu berlatih, yaitu: proses umpan maju dari pola input pelatihan, perhitungan, dan propagasi balik dari error yang terjadi dari penyesuaian nilai pembobot. Pada tahap pelatihan ini merupakan langkah bagaimana suatu jaringan syaraf itu berlatih, yaitu dengan cara melakukan perubahan bobot sambungan, baik bobot sambungan antar input layer dan hidden layer maupun antara hidden layer dan output layer, bila terdapat lebih dari satu hidden layer maka juga terdapat pembobot antar hidden layer itu sendiri. Sedangkan penyelesaian masalah baru akan dilakukan jika proses pelatihan tersebut selesai, fase tersebut adalah proses pemakaian dan testing tentunya dengan menggunakan pembobot yang telah dihasilkan dari proses pelatihan yang telah dilakukan.

Algoritma belajar propagasi balik terdiri dari dua proses, *feedforward* dan *backpropagation* dari errornya. Selama *feedforward* masing-masing unit masukkan menerima (X) atau sinyal masukkan dari luar, kemudian sinyal tersebut disebarkan masing-masing unit pada hidden layer (Z), masing-masing hidden unit menghitung sesuai dengan fungsi aktivasinya. Dan kemudian mengirim sinyal itu ke masing-masing unit pada output layer akan menghitung sesuai dengan fungsi aktivasinya

juga, yang akan menghasilkan sinyal keluaran sebagai respon jaringan dengan adanya pemberian pola input tersebut.

Pada propagasi baliknya, masing-masing output unit dibandingkan dengan hasil perhitungan aktivasi Y dengan nilai target t untuk mendapatkan error, berdasarkan error inilah akan dihitung nilai δ_k , selanjutnya harga error pada output unit akan disebarkan mundur ke masing-masing unit pada hidden layer. Selanjutnya error tersebut digunakan untuk memperbaiki faktor pembobot antara unit output dengan unit hidden demikian selanjutnya dicari error dari keluaran hidden untuk memperbaiki faktor pembobot antara unit input.

Baik tidaknya suatu model NN ditentukan oleh: Pola antar Neuron (arsitektur jaringan). Metode untuk menentukan dan mengubah bobot (disebut metode learning). Fungsi aktivasi NN disebut juga: brain metaphor, computational neuroscience, parallel distributed processing. Dengan kelebihan mampu mengakuisisi pengetahuan walau tidak ada kepastian, mampu melakukan generalisasi dan ekstraksi dari suatu pola data tertentu, NN dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar (self organizing). Perceptron merupakan metode termudah dari *feedforward* neural network. Metode perceptron ini biasanya digunakan untuk mengolah data biner. Persamaan umum dari metode perceptron pada jaringan syaraf *feedforward* ini adalah:

$$\begin{aligned} f(x) &= 1 \text{ jika } w \times x + b > 0 \\ f(x) &= 0 \text{ jika } w \times x + b < 0 \end{aligned} \quad (3.6)$$

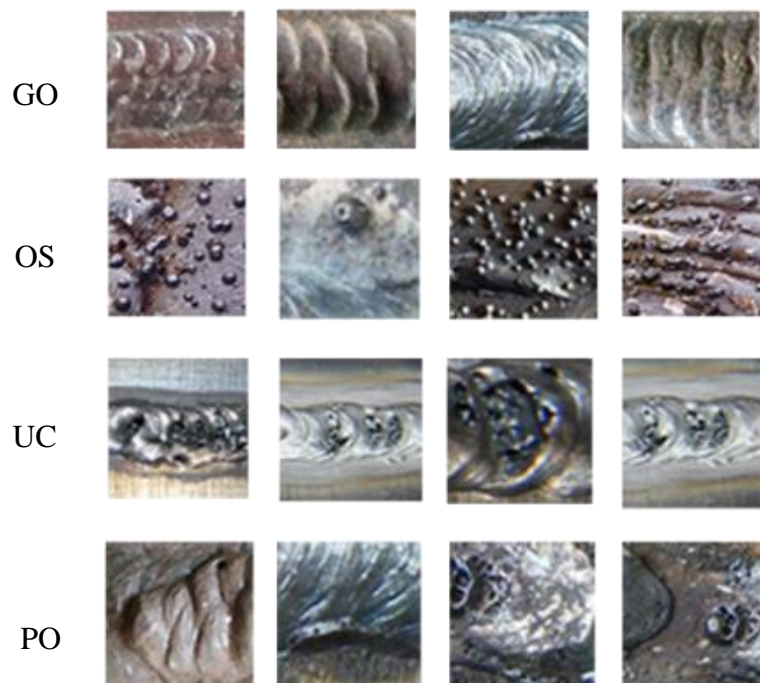
Dengan w adalah bobot, x adalah input sedangkan b adalah bias. Sebenarnya b ini merupakan perkalian antara bias dengan bobot bias tetapi bobot bias ini biasanya diset bernilai 1. Pada perceptron ini biasa digunakan satu atau lebih Neuron, Neuron pada jaringan syaraf tiruan ini berfungsi untuk menghubungkan input dengan output melalui bobot tertentu. Bobot pada jaringan syaraf tiruan merupakan faktor pengali input.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Pembentukan Dataset

Berdasarkan berbagai referensi penelitian seputar aplikasi jaringan syaraf tiruan langkah awal sebelum proses training dimulai adalah pembagian data set untuk keperluan training, pada penelitian ini proporsi data training yang digunakan adalah 80% dari total data, dan untuk keperluan pengujian atau validasi sebanyak 20%. Total data secara keseluruhan adalah 120 data dengan rincian data citra untuk kelas *good* sebanyak 30 data, data citra untuk kelas *over spatter* sebanyak 30 data, data citra untuk kelas *porosity* sebanyak 30 data dan terakhir untuk kelas *undercut* sejumlah 30 data. Semua citra berukuran sama yaitu 320 x 240 piksel, penentuan proporsi data untuk keperluan training dan validasi dilakukan secara acak atau random. Gambar 4.1 adalah sampel dataset untuk masing – masing kelas.



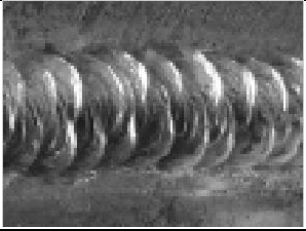
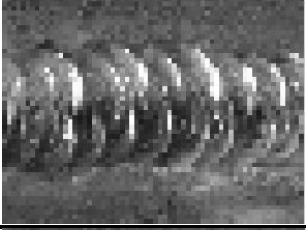
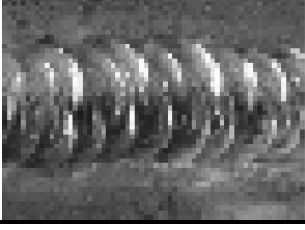
Gambar 4.1 Hasil Pengelasan Kondisi *Good (GO)* , *Over Spatter (OS)*, *Porosity (PO)*, dan *Undercut (UC)*.

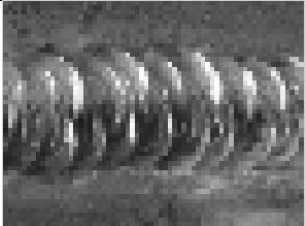
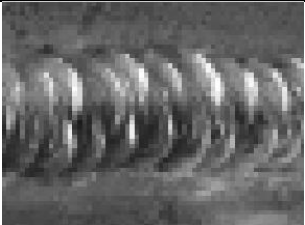
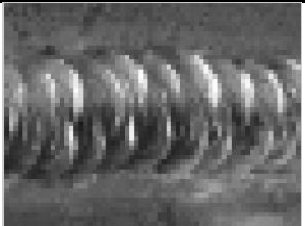
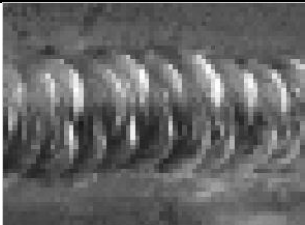
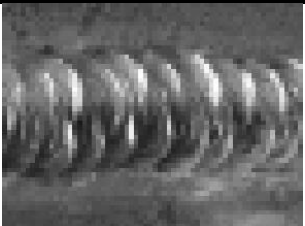
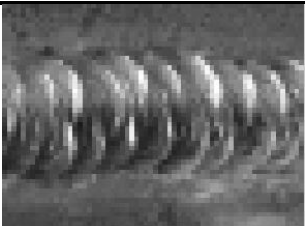
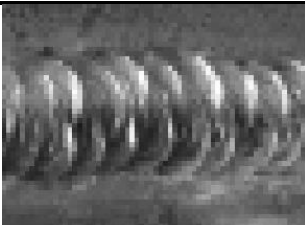
Dari semua total dataset yang digunakan ada 96 (80%) data citra digunakan untuk keperluan training dan 24 (20%) data citra untuk keperluan validasi.

4.2 Konvolusi Citra

Uji coba konvolusi citra dilakukan dengan menggunakan kernel berukuran 5 x 5 sebanyak 10 konvolusi kernel, dengan konstanta deviasi (σ) dibuat berbeda untuk masing – masing kernel. Konvolusi pada dasarnya adalah cara untuk menggabungkan dua array, biasanya untuk ukuran array yang berbeda, tetapi untuk dimensi array yang sama, menghasilkan array ketiga yang mempunyai dimensi yang sama. Konvolusi dapat digunakan dalam image processing untuk menerapkan operator yang mempunyai nilai output dari piksel yang berasal dari kombinasi linear nilai input piksel tertentu. Konvolusi citra mempunyai banyak fungsi seperti yang disebutkan pada bab sebelumnya seperti teknik untuk menghaluskan suatu citra atau memperjelas citra dengan menggantikan nilai piksel dengan sejumlah nilai piksel yang sesuai atau berdekatan dengan piksel aslinya. Dalam bentuk diskrit konvolusi kernel dinyatakan dalam bentuk matriks (umumnya matriks 3x3 / 5x5). Ukuran matriks ini biasanya lebih kecil dari ukuran citra. Setiap elemen matriks disebut koefisien konvolusi. *Feature map* yang terbentuk dari hasil operasi konvolusi dapat dilihat pada Tabel 4.1.

Tabel 4.1 *Feature Maps* untuk Dimensi Kernel 5 x 5 dengan Gaussian Feature pada Citra Pengelasan

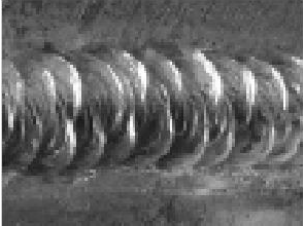
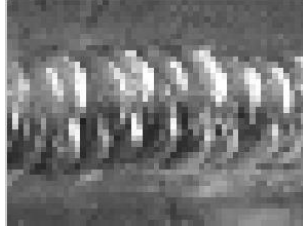
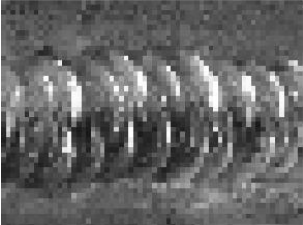
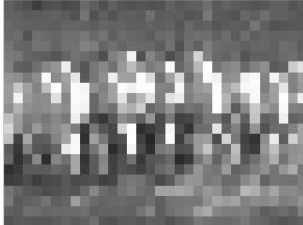
No	Konstanta Deviasi (σ)	<i>Feature Maps</i>
1	10	
2	0.5	
3	0.8	



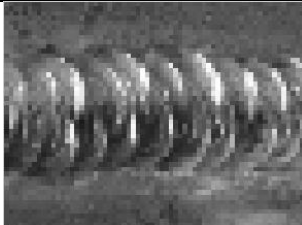
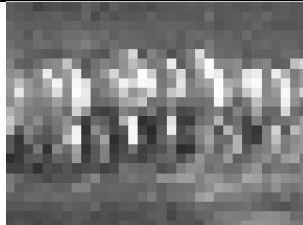
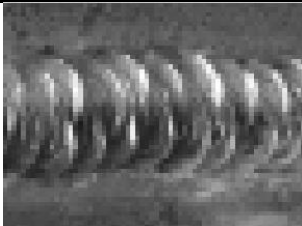
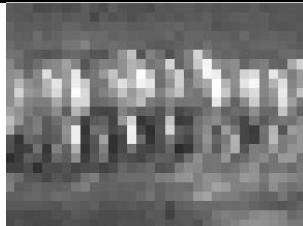
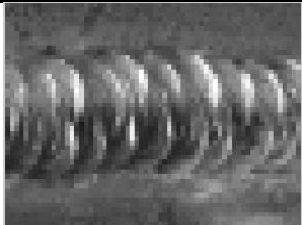
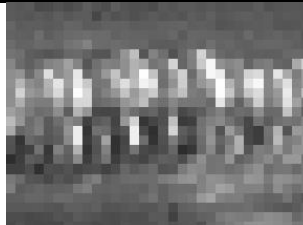
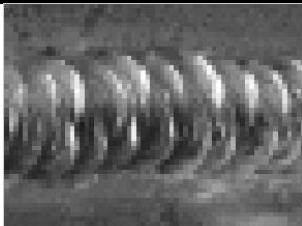
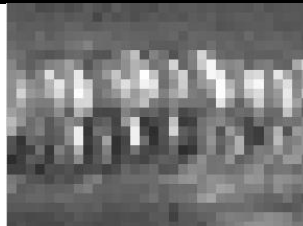
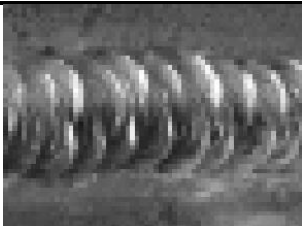
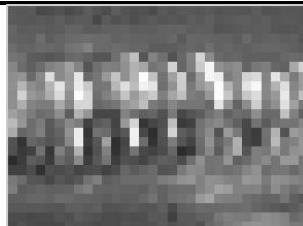
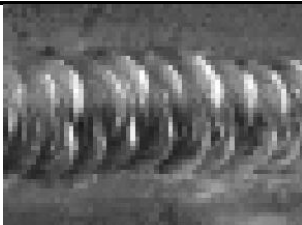
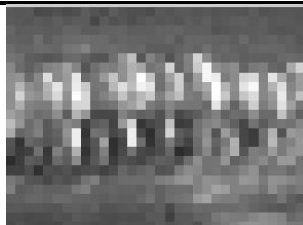
No	Konstanta Deviasi (σ)	Feature Maps
4	1	
5	5	
6	10	
7	20	
8	30	
9	40	
10	45	

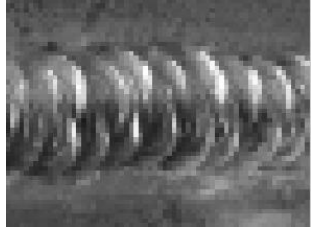
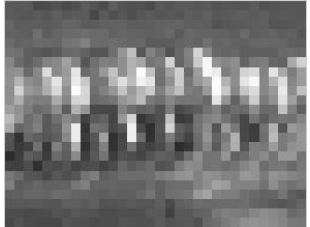
4.3 Hasil ReLu dan Maxpooling Citra

Uji coba proses Relu dan Maxpooling dengan menggunakan gambar hasil konvolusi dengan menggunakan gaussian kernel seperti pada Tabel 4.1 sebelum melewati maxpooling terlebih dahulu harus melewati Relu. ReLu atau Rectified Linear Unit, proses ini sama halnya seperti fungsi aktivasi pada jaringan syaraf tiruan. Dengan tujuan untuk menjaga hasil citra proses konvolusi berada pada domain definit positif. Angka yang dihasilkan harus bernilai positif dikarenakan fungsi aktivasi pada jaringan syaraf tiruan propagasi balik pada penelitian ini menggunakan fungsi sigmoid. Sehingga setiap angka hasil proses konvolusi yang bernilai negatif akan terlebih dahulu melalui proses ReLu yang menjadikan nilai negatif bernilai sama dengan nol. Setelah melewati ReLu barulah tahap selanjutnya adalah Maxpooling yang bertujuan reduksi dimensi. Ukuran citra yang masih besar tentunya menjadi kendala pada saat proses training menggunakan Neural Network oleh sebab itu ukuran citra hasil proses linearisasi (ReLu) dikecilkan melalui proses maxpooling. Maxpooling merupakan proses penyederhanaan matriks yang dilakukan berulang setiap $n \times m$ matriks. Proses tersebut diulangi secara berturut – turut hingga didapatkan citra baru. Dengan catatan jika citra hasil konvolusi merupakan kelipatan ganjil tidak dapat diproses pada tahap maxpooling dengan matriks kelipatan genap. Tabel 4.2 adalah hasil citra baru dari proses ReLu dan Maxpooling.

Tabel 4.2 Hasil ReLu dan Maxpooling pada Feature Maps

	Feature Maps	Maxpooling
1		
2		

No	Feature Maps	Maxpooling
3		
4		
5		
6		
7		
8		
9		

No	Feature Maps	Maxpooling
3		

4.4 Perhitungan Bobot

4.4.1 Analisa Jumlah Neuron Pada Hidden Layer

Analisis jumlah Neuron dilakukan dengan menggunakan arsitektur CNN 2. Pengujian ini sangat penting sebagai dasar keputusan menggunakan 64 Neuron pada layer tersembunyi. Sebelum menentukan 64 Neuron sebagai jumlah Neuron optimum untuk menghasilkan akurasi yang diharapkan penulis melakukan eksperimen secara statistik untuk mengetahui korelasi jumlah Neuron terhadap akurasi yang dihasilkan. Proses perbandingan dilakukan dengan modifikasi hidden Neuron untuk kelipatan 8, 16, 32, dan 64.

1. Training dengan 8 hidden Neuron

Percobaan training pertama mempunyai konfigurasi sebagai berikut:

Jumlah iterasi = 60

Fungsi aktivasi = sigmoid

Learning rate = 0,2

Jumlah hidden Neuron = 8

Dengan konfigurasi parameter tersebut akurasi data validasi yang dihasilkan adalah sebesar 12.20 %. Konfigurasi ini hanya mampu memprediksi 5 dari 41 data validasi dengan benar. Penulis mencoba meningkatkan jumlah Neuron dua kali nilai hidden Neuron sebelumnya.

2. Training dengan 16 hidden Neuron

Dengan parameter jumlah iterasi, learning rate dan fungsi aktivasi yang sama proses iterasi kembali dilakukan dengan perubahan hidden Neuron sebesar 16

Neuron. Setelah 60 iterasi berlangsung nilai rata – rata akurasi yang dihasilkan adalah 41.46%. Akurasi mengalami peningkatan jika dibandingkan dengan kondisi sebelumnya pada saat jumlah Neuron hanya terdiri dari 8 hidden Neuron. Penulis kembali meningkatkan jumlah Neuron dengan harapan mampu memperbaiki tingkat akurasi.

3. Training dengan 32 hidden Neuron

Berikutnya masih dengan parameter yang sama, penambahan jumlah Neuron kali ini tidak memberikan perubahan akurasi yaitu dengan persentase akurasi tetap di angka 41.46% atau data yang berhasil diklasifikasi dengan benar sekitar setengah dari total data validasi.

4. Training dengan 64 hidden Neuron

Peningkatan jumlah Neuron kembali dilakukan tetapi akurasi yang dihasilkan mengalami perubahan yang cukup signifikan yaitu 70.85% sekaligus menjadikan jumlah hidden Neuron ini sebagai struktur CNN untuk kasus cacat las. Pada percobaan terakhir penulis mencoba melipatgandakan jumlah Neuron dengan angka yang cukup besar dan memakan waktu training yang cukup lama.

Tabel 4.3 Korelasi Modifikasi Jumlah Neuron Terhadap Perubahan Akurasi

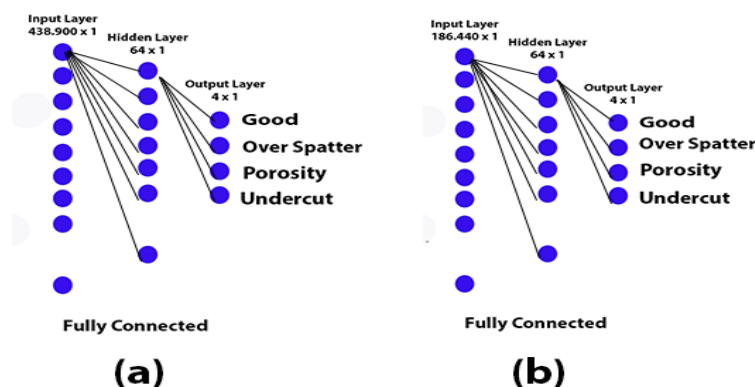
no	jumlah hidden Neuron	akurasi
1	8	12,20%
2	16	41,46%
3	32	41,46%
4	64	70,85%

Sesuai Tabel 4.3 yang menjelaskan rekapitulasi hasil akurasi secara statistik untuk setiap modifikasi jumlah hidden Neuron maka penulis memutuskan untuk memilih 64 Neuron sebagai struktur utama layer tersembunyi. Belum adanya referensi yang mendukung yang membahas tentang korelasi jumlah hidden layer

berdasarkan aplikasinya membuat percobaan trial error seperti ini perlu dilakukan untuk mendapatkan akurasi yang optimum.

4.4.2 Analisa Arsitektur CNN 1 dan CNN2.

Pada penelitian ini arsitektur CNN yang digunakan terdapat dua jenis seperti yang ditunjukkan pada Gambar 3.2 dan Gambar 3.3. yang membedakan dari keduanya selain tahapan proses ekstraksi *feature* adalah jumlah vektor input untuk proses fully connected-nya yang terdiri 438.900 untuk arsitektur CNN 1 dan 180.960 untuk arsitektur CNN 2. Sedangkan jumlah Neuron pada hidden layer mempunyai jumlah yang sama yaitu 64 Neuron sesuai hasil dengan hasil percobaan pada bab 4.4.1, semua Neuron setiap layer akan terhubung secara penuh ke Neuron layer berikutnya. Selain itu terdapat parameter utama yang digunakan pada Neural Network propagasi balik dengan konsep gradient descent yaitu menggunakan α laju pembelajaran atau lebih dikenal dengan *learning rate* penentuan parameter learning rate yang sesuai akan mempengaruhi perubahan bobot pada setiap layer yang mampu mengoptimalkan hasil kalkulasi terhadap target. Sudah dijelaskan pada Subbab 3.1 bahwa vektor output yang dimiliki terdiri dari 4 yaitu *good*, *undercut*, *over spatter*, dan *porosity*. Struktur jaringan yang terbentuk untuk arsitektur CNN 1 dan CNN 2 dapat dilihat pada Gambar 4.2, serta nilai bobot untuk input layer dan hidden layer dapat dilihat pada Tabel 4.4 untuk arsitektur CNN 1 dan Tabel 4.5 untuk arsitektur CNN 2. Sedangkan bobot hidden layer dan output layer dapat dilihat pada Tabel 4.6.



Gambar 4.2 Jaringan Neural Network yang Terbentuk, (a) Arsitektur CNN 1, (b) Arsitektur CNN 2.

Tabel 4.4 Bobot Training Input Layer dan Hidden Layer untuk Arsitektur CNN 1.

vektor input	Neuron Hidden Layer														
		1	2	3	4	5	6	59	60	61	62	63	64
	1	0,0330	0,0279	0,0283	-0,0587	-0,0244	-0,0503	-0,0387	-0,0204	0,1103	-0,0007	-0,0733	-0,0367
	2	-0,0316	0,0275	-0,0914	-0,0183	0,0189	0,0112	0,0544	-0,0828	0,0097	-0,0202	0,0607	0,0210
	3	0,0666	-0,0187	-0,0099	0,0070	0,0236	-0,0032	-0,0129	-0,0245	0,0488	0,0026	-0,0231	-0,0267
	4	-0,0037	0,0000	-0,0585	0,0341	-0,0089	-0,0648	0,0403	-0,0156	0,0627	0,0127	0,0102	-0,0474
	5	0,0143	0,0035	0,0473	0,0443	-0,0509	-0,0028	-0,0417	0,0389	0,0519	0,0171	0,0012	-0,0372
	6	0,0535	-0,0177	-0,0164	0,0546	-0,0352	-0,0392	-0,0759	0,0354	-0,0311	0,0453	-0,0011	-0,0283
	7	-0,0007	0,0064	-0,0072	-0,0071	0,0190	0,0373	-0,0081	0,0160	0,0088	-0,0108	-0,0174	-0,0364
	8	-0,0113	0,0257	-0,0347	0,0129	0,0239	0,0340	0,0062	-0,0514	0,0390	0,0438	-0,0331	-0,0532

	438895	-0,0119	-0,0184	0,0330	0,0578	-0,0660	-0,0593	-0,0438	0,0148	0,0501	-0,0078	0,0198	-0,0666
	438896	0,0510	0,0307	0,0064	0,0123	0,0161	0,0379	0,0125	-0,0436	0,0223	0,0314	0,0543	0,0373
	438897	0,0488	-0,0021	-0,0186	-0,0344	0,0130	0,0218	-0,0604	-0,0425	0,0355	0,0114	-0,0145	0,0108
	438898	-0,0157	-0,0124	0,0235	0,0081	0,0149	0,0318	0,0100	0,0559	0,0505	0,0696	-0,0040	-0,0506
	438899	0,0111	0,0458	-0,0373	0,0390	0,0301	0,0392	0,0490	-0,0283	-0,0171	0,0065	0,0301	-0,0006
	180900	-0,0227	0,0243	-0,0311	0,0056	-0,0402	-0,0097	0,0473	-0,0749	0,0982	0,0025	0,0164	-0,0030

Tabel 4.5 Bobot Training Input Layer dan Hidden Layer untuk Arsitektur CNN 2.

vektor input	Neuron Hidden Layer														
		1	2	3	4	5	6	59	60	61	62	63	64
	1	0,0259	0,0445	0,0426	-0,0115	0,0002	0,0332	-0,0443	-0,0125	-0,0480	-0,0430	-0,0129	0,0346
	2	-0,0133	0,0055	-0,0503	-0,0324	-0,0176	0,0126	-0,0261	0,0062	0,0125	-0,0320	0,0225	0,0282
	3	-0,0195	0,0239	0,0393	-0,0138	0,0333	0,0001	0,0182	-0,0074	-0,0152	-0,0020	0,0005	0,0008
	4	-0,0077	0,0463	-0,0349	-0,0123	0,0369	0,0016	-0,0172	0,0322	0,0412	-0,0229	-0,0466	-0,0194
	5	-0,0087	0,0016	-0,0500	-0,0250	-0,0246	0,0340	-0,0298	-0,0204	-0,0437	-0,0200	-0,0324	0,0345
	6	-0,0124	-0,0324	-0,0109	0,0050	0,0189	-0,0032	0,0056	-0,0122	-0,0153	0,0204	0,0203	-0,0081
	7	-0,0245	-0,0358	-0,0451	-0,0348	0,0148	0,0269	-0,0312	0,0282	0,0018	0,0015	0,0114	-0,0422
	8	0,0110	-0,0066	-0,0309	0,0460	-0,0171	-0,0415	0,0034	0,0386	0,0025	0,0181	0,0270	-0,0231

	186435	0,0085	-0,0040	0,0076	0,0069	0,0141	0,0474	0,0138	0,0275	-0,0497	-0,0437	0,0341	0,0199
	186436	0,0435	0,0404	-0,0522	-0,0283	-0,0108	-0,0297	0,0265	0,0091	0,0271	0,0037	0,0050	0,0149
	186437	0,0060	0,0050	-0,0542	0,0296	-0,0148	-0,0072	-0,0096	-0,0070	-0,0362	0,0386	0,0429	-0,0250
	186438	-0,0379	0,0287	0,0123	-0,0477	0,0322	0,0050	-0,0171	0,0444	-0,0370	-0,0271	0,0309	-0,0282
	186439	0,0091	0,0388	-0,0017	-0,0386	0,0072	-0,0128	0,0054	0,0012	-0,0285	0,0296	-0,0044	-0,0290
	186440	-0,0355	0,0214	0,0385	-0,0207	-0,0452	0,0065	-0,0133	-0,0153	-0,0225	-0,0036	-0,0001	0,0340

Tabel 4.6 Bobot Training Hidden Layer dan Output Layer untuk Arsitektur CNN 1 dan 2.

	Vektor Output CNN1			
	1	2	3	4
1	-0,1386	-0,1100	0,2531	0,2384
2	0,1865	0,0398	-0,1440	-0,0758
3	-0,3565	-0,2189	0,3600	0,2061
4	0,1744	-0,2289	0,2295	0,1014
5	-0,2060	0,0024	-0,2651	0,0941
6	-0,2590	-0,1661	0,1555	0,0186
7	0,0973	0,1630	0,0459	-0,1104
8	0,0357	0,1975	-0,0104	0,2467
9	-0,2187	-0,3309	0,0521	0,2460
10	-0,0715	-0,3276	-0,0761	-0,1970
11	0,1850	-0,0197	0,1077	-0,2378
12	0,0516	-0,1693	0,3202	0,1721
13	-0,2736	-0,1065	0,2750	0,3980
14	0,1873	0,1032	-0,1134	0,1425
15	0,2884	0,3499	-0,0212	-0,1669
16	-0,1389	0,1250	-0,1259	0,0627
17	-0,2779	0,1209	0,0701	-0,2930
18	0,1878	-0,0582	-0,0237	-0,2069
19	-0,4418	0,2648	0,1655	0,1176
...
...
52	-0,1136	-0,2204	-0,0430	0,0164
53	-0,1989	-0,1916	0,0810	-0,1193
54	0,1316	-0,1413	0,0679	0,2290
55	-0,1719	-0,2388	0,0568	-0,0056
56	0,1007	0,2762	0,2956	0,0428
57	0,2174	-0,1202	-0,2066	0,2072
58	0,0774	-0,0895	0,2617	-0,2921
59	0,2256	-0,1590	-0,0293	-0,0131
60	0,0246	-0,1282	0,2812	0,3183
61	0,2744	0,1802	0,2733	-0,0443
62	0,2079	-0,2008	0,1543	0,1842
63	-0,0541	-0,0538	-0,1605	-0,1729
64	0,0928	-0,3899	0,0787	0,0745

	Vektor Output CNN2			
	1	2	3	4
1	0,1319	-0,2151	0,2673	-0,28
2	-	0,1279	0,2587	0,021
3	0,2282	-0,2083	0,0881	-0,2487
4	-	0,2828	-0,0684	-0,1544
5	-	0,2321	0,1201	-0,2234
6	-	0,2924	-0,112	-0,1475
7	0,0561	0,2692	-0,0036	0,2122
8	-	0,1618	-0,2683	0,0573
9	0,2013	-0,0919	-0,2264	0,0308
10	0,0093	-0,2978	0,2781	-0,0732
11	0,1987	0,023	0,107	-0,0896
12	0,0128	0,071	0,1763	-0,207
13	-	0,1509	0,1009	0,2901
14	-	0,1626	0,0448	-0,2054
15	0,0995	-0,2986	0,1111	-0,0735
16	-	0,1831	0,0093	0,1827
17	0,0976	-0,1045	0,1324	-0,2195
18	0,3017	0,1382	0,1865	0,1158
19	0,043	0,1866	0,0111	-0,0259
...
...
...
52	0,0003	-0,219	0,0876	0,1044
53	0,1391	0,0541	0,118	-0,2122
54	0,1817	0,0414	-0,0396	-0,0167
55	0,2317	0,1722	0,1004	-0,2897
56	-	0,2166	-0,0664	0,2954
57	0,2765	-0,2827	0,2853	0,2492
58	-0,188	0,1856	0,1025	-0,1607
59	0,3225	-0,2908	-0,2431	-0,2412
60	0,2312	0,1446	-0,2465	0,2168
61	0,1897	-0,0164	0,2848	-0,1273
62	-	0,0559	0,281	-0,0822
63	0,0091	0,0933	-0,1355	-0,1418
64	-	0,138	-0,1164	0,1935

4.5 Proses Training dan Validasi

Langkah awal untuk menggunakan spyder untuk aplikasi pengolahan citra berbasis konvolusi yaitu import keras.io, keras adalah library untuk jaringan syaraf tiruan tingkat lanjut dengan skala komputasi yang besar (deep learning). Import semua komponen seperti fungsi *dense* yang merupakan variable deklarasi untuk penentuan jumlah Neuron pada hidden layer, *activation* untuk menentukan fungsi aktivasi yang digunakan. Pada penelitian ini dua fungsi aktivasi yang digunakan adalah ReLu dan sigmoid. *Flatten* atau pada bab sebelumnya disebut flattening merupa matriks berdimensi $n \times m$ menjadi komponen bilangan individu sebagai vektor input. Fasilitas Convolution 2D sebagai proses konvolusi citra dengan berbagai kernel didalamnya serta Maxpool 2D untuk proses subsampling dimana matrix maxpooling dapat diatur sesuai kebutuhan.

Variabel data training dan data testing yang digunakan dapat dilihat pada Tabel 4.7, meliputi jumlah data yang akan training dan testing, besar resolusi, dan channel warna.

Tabel 4.7 Parameter Input Dataset

	Data Training	Data Testing
Jumlah Data	96	24
Resolusi	320 x 240 Pixel	320 x 240 Pixel
Ruang Warna	Grayscale	Grayscale

Data testing adalah citra yang digunakan untuk proses validasi dengan keterangan pada kolom size (24, 320, 240, 1). Maksud dari kolom size tersebut adalah total data citra yang digunakan untuk keperluan validasi adalah 24 sesuai dengan proporsi data yang sudah dijelaskan sebelumnya sebanyak 20 % dari total 120 data. 320,240,1 adalah ukuran citra grayscale. Data training merupakan citra yang dipakai untuk proses training dengan keterangan pada kolom size (96, 320, 240, 1). Sebanyak 96 data citra digunakan untuk keperluan training.

Seperti yang telah dijelaskan pada bab sebelumnya tentang Neural Network propagasi balik terdapat beberapa parameter yang harus ditentukan terlebih dahulu diantaranya jumlah iterasi, jumlah iterasi yang digunakan pada penelitian ini adalah

60. Menggunakan single hidden layer *backpropagation* neural network dengan jumlah Neuron pada hidden layer sebanyak 64 Neuron. Serta fungsi aktivasi yang digunakan yaitu sigmoid. Jumlah vektor input dari proses subsampling terakhir adalah 57 x 77 setelah melalui proses flattening sehingga vektor input menjadi 438.900 untuk arsitektur CNN 1 dan 186.440 untuk arsitektur CNN 2.

Vektor output pada penelitian ini berupa bilangan biner dengan 4 vektor output sesuai dengan klasifikasi citra las yaitu *good*, *overs patter*, *porosity* dan *undercut*. Klasifikasi output jaringan syaraf tiruan lebih lanjut dijelaskan pada Tabel 4.8.

Tabel 4.8 Nilai Biner Vektor Output untuk Masing – Masing Kategori

klasifikasi	Vektor output (biner)	Klasifikasi	Vektor output (biner)
<i>Good</i>	[1 0 0 0]	<i>Porosity</i>	[0 0 1 0]
<i>Over spatter</i>	[0 1 0 0]	<i>Undercut</i>	[0 0 0 1]

Vektor output berupa bilangan biner dipilih untuk memudahkan proses analisis data. Klasifikasi berupa bilangan biner dipakai untuk proses labeling pada vektor output baik untuk label testing label training.

Parameter yang digunakan untuk evaluasi hasil training adalah MSE dan akurasi. Nilai MSE merupakan kuadrat dari jumlah selisih antara target (t) dengan keluaran hasil prediksi y_k disetiap iterasi (i) dibagi dengan jumlah iterasi maksimum (n).

$$MSE = \frac{\sum (t_i - Y_{ki})^2}{n} \quad (4.1)$$

Mean Squared Error (MSE) adalah metode untuk mengevaluasi metode peramalan. Masing-masing kesalahan atau sisa dikuadratkan. Kemudian dijumlahkan dan ditambahkan dengan jumlah observasi. Pendekatan ini mengatur kesalahan peramalan yang besar karena kesalahan-kesalahan itu dikuadratkan.

Metode itu menghasilkan kesalahan-kesalahan sedang yang kemungkinan lebih baik untuk kesalahan kecil, tetapi kadang menghasilkan perbedaan yang besar. MSE sangat sensitif terhadap *outlier*. Karena dihitung nilai kuadratnya, *error outlier* akan diberikan bobot yang sangat besar dan membuat nilai MSE semakin besar pula. MSE sangat baik dalam memberikan gambaran terhadap seberapa konsisten model yang dibangun. Dengan meminimalkan nilai MSE, berarti meminimalkan varian model. Model yang memiliki varian kecil mampu memberikan hasil yang relatif lebih konsisten untuk seluruh data input dibandingkan dengan model dengan varian besar (MSE besar). Dalam kasus klasifikasi biner, dimana hanya terdapat dua kelas dengan label kelas 0 dan 1, tidak ada perbedaan dalam penggunaan MSE. Hal ini karena nilai *error* hanya mempunyai dua kemungkinan, 0 jika prediksi benar dan 1 jika prediksi kelas berbeda dengan kelas sebenarnya. Hasil kuadrat ataupun absolut dari *error* tersebut akan sama sehingga nilai MSE pun akan identik.

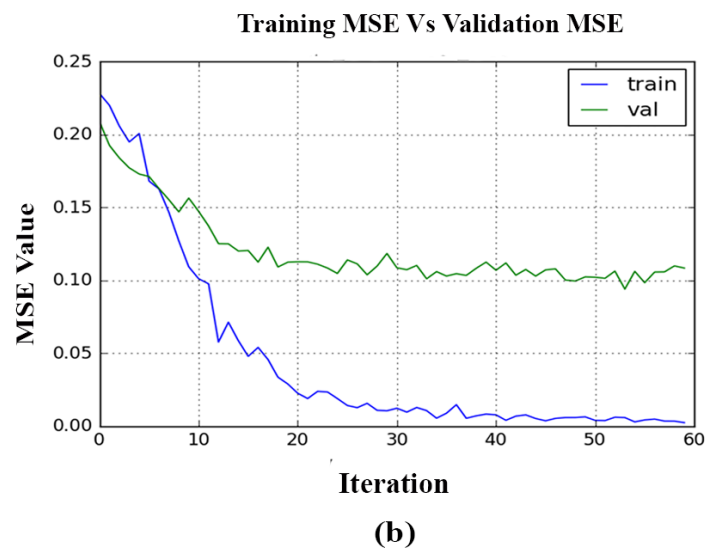
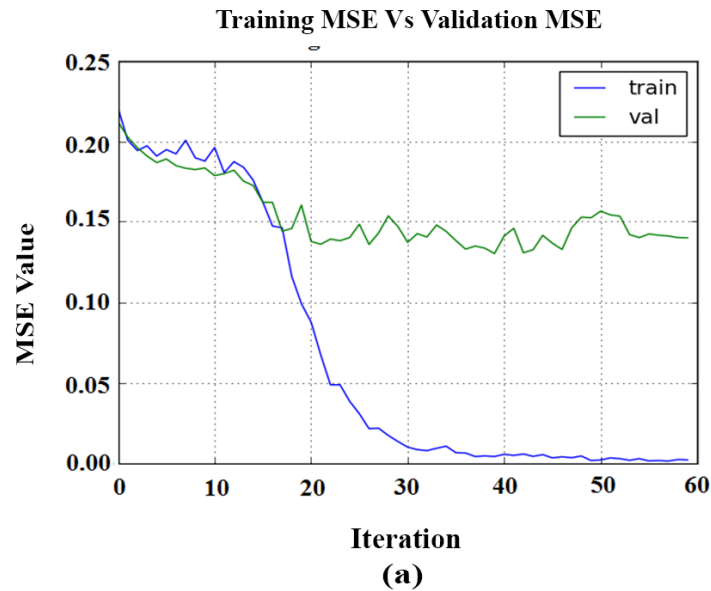
Alternatif yang lain adalah dengan menarik akar kuadrat MSE, atau yang biasa disebut dengan *Root Mean Squared Error* (RMSE). RMSE menjadi alternatif yang lebih intuitif dibandingkan MSE karena memiliki skala pengukuran yang sama dengan data yang sedang dievaluasi. Sebagai contoh, dua kali nilai RMSE artinya model memiliki *error* dua kali lebih besar dari sebelumnya. Sedangkan dua kali nilai MSE tidak berarti demikian. Jika MSE dapat dianalogikan sebagai varian, maka RMSE dapat dianalogikan sebagai standar deviasi.

Sedangkan untuk menentukan persentase akurasi dapat dihitung dengan jalan jumlah prediksi benar (y_b) dibagi dengan total data training (m) dikalikan dengan 100%.

$$\text{Accuracy} = \frac{\sum y_b}{m} \cdot 100 \% \quad (4.2)$$

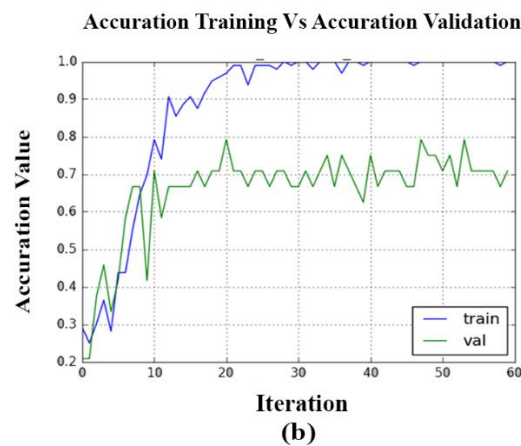
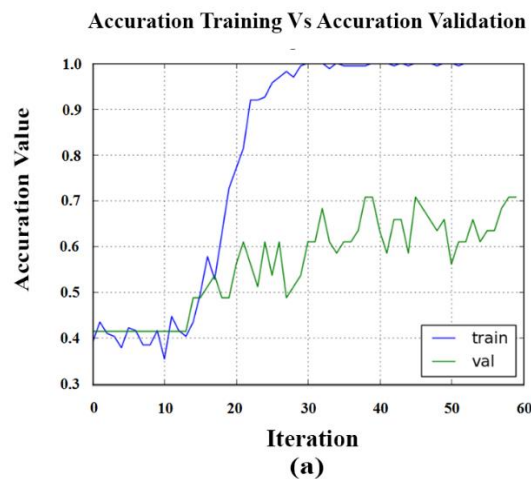
Proses validasi bertujuan untuk menentukan performa CNN terhadap data citra yang belum diketahui sebelumnya, 24 data citra digunakan untuk proses validasi ini, sama halnya dengan proses training dua parameter yang akan dikaji lebih lanjut adalah MSE dan akurasi.

MSE yang didapatkan setelah 60 kali iterasi untuk 96 data training adalah 0,0681 dan akurasi rata – rata yang dihasilkan dari iterasi awal hingga akhir adalah sebesar 80,35%. Sedangkan untuk 24 data validasi memiliki nilai MSE sebesar 0,1399 dan akurasi rata – rata berada dikisaran 70,73%.



Gambar 4.3 Hasil MSE Untuk Data Training (Biru) Dan Data Testing (Hijau), (a) Arsitektur 1, (b) Arsitektur 2.

Dari Gambar 4.3 Nilai rata – rata kesalahan untuk data training dimulai di kisaran 0,2 dan bergerak konstan mendekati nilai 0.00 setelah iterasi ke-30 untuk kedua arsitektur, rata – rata kuadrat kesalahan disetiap iterasi adalah 0,0681. Untuk data validasi, nilai MSE dimulai di kisaran 0,2 dan stabil pada saat iterasi ke 20. Respon MSE untuk data validasi setelah iterasi ke-20 berada diantara 0,13 hingga 0,16 dengan rata – rata kuadrat kesalahan yaitu 0,145 untuk arsitektur CNN1, sedangkan untuk arsitektur CNN 2 setelah iterasi ke-20 berada diantara 0.8 – 0.13 dengan rata-rata kuadrat kesalahan adalah 0.12. Untuk tingkat akurasi dari jumlah dataset yang dapat dikenali dengan data input yang digunakan sebagai testing dapat dilihat pada Gambar 4.4.



Gambar 4.4 Hasil Akurasi Untuk Data Training (Biru) Dan Data Testing (Hijau) ,
(a) Arsitektur 1, (b) Arsitektur 2.

Gambar 4.4a menunjukkan akurasi data training dimulai pada angka 0,4 atau sekitar 40 persen untuk arsitektur CNN 1 dan secara bertahap meningkat hingga pada iterasi ke-30 mulai stabil di angka 1,0 atau mendekati 100 % dengan rata – rata akurasi mulai dari iterasi pertama hingga terakhir yaitu 80,35 %. Maksud dari 80.35 % adalah dari total keseluruhan data training yang berjumlah 96 data citra, sebanyak 77 data dapat diklasifikasikan dengan benar.

Grafik warna hijau pada Gambar 4.4a menunjukkan respon akurasi untuk data validasi, sama seperti respon untuk data training pada iterasi pertama persentase akurasi data testing yang ditunjukkan adalah dikisaran 40% dan terus meningkat hingga pada iterasi ke-39 mencapai 70,73 persen. Dan pada akhir iterasi tingkat akurasi turun mencapai 66.05 %, Sehingga performa CNN pada penelitian ini mampu mengenali data citra las dengan tingkat kebenaran sejumlah 16 data dari total 24 data yang diujikan. Dapat dilihat dari grafik a bahwa respon akurasi masih sangat fluktuatif antara data yang sebelumnya dengan data yang sedang diuji saat ini.

Sedangkan untuk untuk arsitektur CNN 2 seperti pada Gambar 4.4b akurasi data training diawali pada kisaran angka 0,3 atau sekitar 30 persen untuk dan secara bertahap meningkat hingga pada iterasi ke-10 mulai stabil di angka 1,0 atau mendekati 100 % dengan rata – rata akurasi mulai dari iterasi pertama hingga terakhir yaitu 90,35 %, detail nilai akurasi dan validasi untuk arsitektur CNN 2 dapat dilihat pada Tabel 4. 9. Dari hasil training, data training yang dapat dikenali sebanyak 87 data. Sedangkan untuk respon akurasi untuk data validasi, iterasi pertama persentase akurasi data testing yang ditunjukkan adalah dikisaran 10% dan terus meningkat hingga pada iterasi ke-10 mencapai 70,83 persen. Dan pada akhir iterasi tingkat akurasi mencapai 70.85 %. Sehingga mampu mengenali data citra las dengan tingkat kebenaran sejumlah 17 data dari total 24 data yang diujikan. Dapat dilihat dari grafik a bahwa respon akurasi lebih stabil dibandingkan dengan arsitektur CNN 1. Hasil perbandingan nilai prediksi ketika menggunakan arsitektur CNN 1 dan CNN2 dapat dilihat pada Tabel 4.10.

Tabel 4. 9 Respon Akurasi Data Validasi

No	Train acc	Train MSE	Test acc	Test MSE
1	0,291666669	0,227878845	0,20833334	0,208565224
2	0,250000007	0,219856264	0,20833334	0,192341257
3	0,302083338	0,20557793	0,375000002	0,183729668
4	0,364583342	0,194711972	0,458333333	0,176944664
5	0,281250005	0,200506627	0,333333337	0,172722672
6	0,437500008	0,167857418	0,416666679	0,171002382
7	0,437500005	0,162433949	0,583333343	0,162750637
8	0,552083337	0,14644323	0,666666672	0,15508761
9	0,645833336	0,126780385	0,666666667	0,146729564
10	0,697916666	0,109093562	0,416666672	0,156258207
11	0,791666667	0,100905356	0,708333333	0,147094547
12	0,739583334	0,097427129	0,583333338	0,137108641
13	0,906249996	0,057472883	0,666666657	0,124957022
14	0,854166669	0,071120749	0,666666667	0,12484545
15	0,885416658	0,058446559	0,666666667	0,119923074
16	0,906249984	0,047680361	0,666666657	0,120268122
17	0,874999995	0,05381709	0,708333333	0,112303792
18	0,916666653	0,045460246	0,666666657	0,122545157
19	0,94791666	0,033381814	0,708333333	0,108944765
20	0,958333323	0,028687215	0,708333333	0,112286936
21	0,968749993	0,022334288	0,791666652	0,112545452
22	0,989583331	0,018734485	0,708333318	0,11249282
23	0,989583331	0,02365106	0,708333318	0,110899743
24	0,937499991	0,023217958	0,666666657	0,108232152
25	0,989583331	0,018720802	0,708333318	0,104520461
26	0,989583332	0,013969343	0,708333323	0,113827175
27	0,989583331	0,012480981	0,666666667	0,111120017
28	0,979166662	0,015465085	0,708333333	0,103588437
29	1	0,010670472	0,708333318	0,109573642
30	0,989583331	0,010344372	0,666666667	0,118253184
31	1	0,012073498	0,666666667	0,108413558
32	1	0,009433967	0,708333333	0,107121243
33	0,979166662	0,012577702	0,666666667	0,110023762
34	1	0,010454473	0,708333318	0,100870547
35	1	0,005263709	0,75	0,105794076

No	Train acc	Train MSE	Test acc	Test MSE
36	1	0,008645142	0,666666667	0,102754252
37	0,968749999	0,014542955	0,75	0,104363961
38	1	0,005204978	0,708333318	0,103198676
39	1	0,006860542	0,666666667	0,108050381
40	0,989583331	0,008058509	0,625000015	0,112335211
41	1	0,007542656	0,75	0,10673824
42	1	0,003805942	0,666666667	0,111763353
43	1	0,006668504	0,708333333	0,103335123
44	1	0,007521117	0,708333333	0,107269434
45	1	0,005046136	0,708333318	0,102745715
46	1	0,003466949	0,666666667	0,106976577
47	0,989583331	0,005120282	0,666666667	0,10769004
48	1	0,005707202	0,791666652	0,100016195
49	1	0,005723289	0,75	0,099355755
50	1	0,006162551	0,749999985	0,102178056
51	1	0,003749673	0,708333318	0,101941804
52	1	0,003610786	0,749999985	0,101235138
53	1	0,005917009	0,666666667	0,106155433
54	1	0,00568903	0,791666652	0,093782624
55	1	0,002722324	0,708333333	0,106008299
56	1	0,004042949	0,708333318	0,098198072
57	1	0,00463541	0,708333333	0,105481127
58	1	0,003272885	0,708333333	0,105624907
59	0,989583331	0,003230709	0,666666667	0,109702287
60	1	0,00220323	0,708333333	0,108219777

Dari Tabel 4.10 dapat diambil kesimpulan bahwa selisih prediksi citra input untuk arsitektur CNN1 dan CNN2 hanya 1 yaitu untuk arsitektur 1 sebanyak 16 data yang dapat dikenali sedangkan arsitektur 2 sebanyak 17 data yang dapat dikenali. Oleh karena itu arsitektur CNN terbaik yang digunakan adalah arsitektur CNN 2. Adapun distribusi sebaran data untuk empat kategori las terdapat pada Tabel 4.11.

Tabel 4.10 Hasil Prediksi Untuk 24 Data Validasi

No	Nilai Benar	Nilai Prediksi CNN1	Nilai Prediksi CNN 2	Nilai Kebenaran CNN 1	Nilai Kebenaran CNN 2
1	[1 0 0 0]	[1 0 0 0]	[1 0 0 0]	1	1
2	[0 0 1 0]	[0 0 1 0]	[0 0 1 0]	1	1
3	[1 0 0 0]	[0 0 1 0]	[0 0 1 0]	0	0
4	[0 0 0 1]	[0 0 0 1]	[0 0 0 1]	1	1
5	[0 1 0 0]	[0 1 0 0]	[0 1 0 0]	1	1
6	[0 0 0 1]	[1 0 0 0]	[1 0 0 0]	0	0
7	[1 0 0 0]	[0 0 0 1]	[1 0 0 0]	0	1
8	[0 1 0 0]	[0 1 0 0]	[0 1 0 0]	1	1
9	[0 0 1 0]	[0 0 1 0]	[0 0 1 0]	1	1
10	[1 0 0 0]	[1 0 0 0]	[1 0 0 0]	1	1
11	[0 0 1 0]	[0 0 1 0]	[0 0 1 0]	1	1
12	[0 0 1 0]	[0 0 1 0]	[0 0 1 0]	1	1
13	[0 0 1 0]	[0 0 0 1]	[0 0 0 1]	0	0
14	[0 1 0 0]	[0 0 1 0]	[0 0 1 0]	0	0
15	[0 0 0 1]	[0 0 0 1]	[0 0 0 1]	1	1
16	[0 0 1 0]	[0 0 1 0]	[0 0 1 0]	1	1
17	[0 0 0 1]	[0 0 1 0]	[0 0 1 0]	0	0
18	[0 1 0 0]	[0 1 0 0]	[0 1 0 0]	1	1
19	[0 1 0 0]	[0 1 0 0]	[0 1 0 0]	1	1
20	[1 0 0 0]	[1 0 0 0]	[1 0 0 0]	1	1
21	[0 0 1 0]	[0 0 1 0]	[0 0 1 0]	1	1
22	[0 0 0 1]	[0 0 0 1]	[0 0 0 1]	1	1
23	[0 0 1 0]	[1 0 0 0]	[1 0 0 0]	0	0
24	[0 0 0 1]	[0 0 1 0]	[0 0 1 0]	0	0
	Total benar			16	17
	Persentase Akurasi			66.66	70,83333333

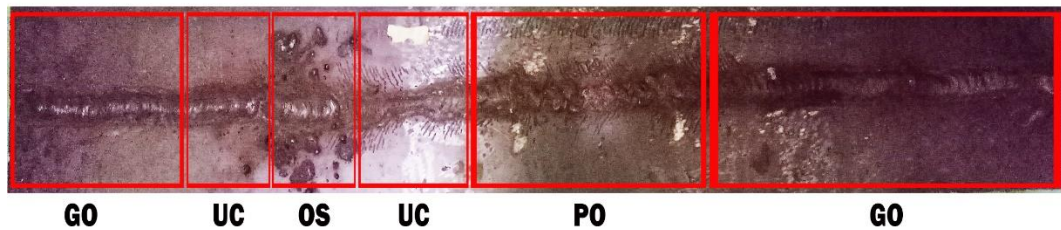
Tabel 4.11 Distribusi sebaran data citra

Kategori	Total Data	CNN 1		CNN 2	
		Benar	Salah	Benar	Salah
<i>Good</i>	5	3	2	4	1
<i>Over spatter</i>	5	4	1	4	1
<i>Porosity</i>	8	6	2	6	2
<i>Undercut</i>	6	3	3	3	3
Total	24	16	8	17	7

Tabel 4.11 menjelaskan distribusi data validasi untuk masing – masing kategori serta rekapitulasi hasil prediksi CNN untuk kondisi benar dan salah. Secara statistik dapat diketahui bahwa arsitektur CNN 1 mampu memprediksi kondisi good dengan jumlah benar lebih dominan jika dibandingkan dengan kondisi yang lain, disusul *porosity*, *overs patter* dan yang terakhir *undercut*.

4.6 Pengujian Real Time Untuk Hasil Pengelasan








Tahapan pengujian ini dilakukan secara relatime dengan citra diambil langsung melalui kamera dengan posisi kamera bergerak dan hasil pengelasan didekatkan terhadap lensa kamera. Resolusi citra yang diambil adalah berukuran 320 x 240 dengan ruang warna R, G, dan B. Seperti yang sudah dijelaskan sebelumnya bahwa citra yang akan diproses harus citra dengan ruang warna grayscale sehingga harus diubah terlebih dahulu. Sama seperti saat melakukan training citra, input juga akan melalui proses konvolusi dan subsampling hingga proses flattening. Bobot yang telah tersimpan dari proses training dengan akurasi optimum 70.85% akan digunakan untuk mengklasifikasi empat sampel hasil pengelasan yaitu *good*, *porosity*, *undercut*, dan *over spatter*. Uji coba dilakukan dengan menyimpan semua citra yang ditangkap oleh kamera secara realtime yang kemudian akan dipisahkan antara hasil klasifikasi yang benar dan salah, pengujian dilakukan pada satu benda kerja yang mempunyai beberapa kategori cacat seperti yang ditunjukkan pada Gambar 4.5.






Gambar 4.5 Contoh hasil pengelasan yang akan diuji. GO (Good), UC(Undercut), OS (Overspater), PO (Porosity).

Dari hasil pengujian yang dilakukan terdapat 217 citra yang tersimpan dengan masing-masing citra diberi label sesuai dengan hasil klasifikasinya. Untuk hasil pengelasan yang baik, arsitektur CNN yang digunakan sudah dapat mengenali hasil pengelasan yang baik sejumlah 86 data dari total 108 data, beberapa hasil klasifikasi seperti yang ditunjukkan pada Tabel 4.12.

Tabel 4.12 Hasil Klasifikasi Pengelasan yang Baik Secara *Real Time*.


No	Hasil Klasifikasi	Keterangan
1		Benar
2		Benar
3		Benar
4		Benar
5		Benar
6		Benar
7		Benar

No	Hasil Klasifikasi	Keterangan
8		Benar
9		Salah
10		Salah
11		Salah
12		Benar
13		Benar
14		Benar
15		Benar
16		Benar

Uji coba selanjutnya bertujuan untuk mengklasifikasikan hasil pengelasan yang mempunyai cacat *Over Spatter*, untuk kategori cacat ini jumlah data yang dihasilkan terdiri 21 citra, sama halnya dengan pengujian sebelumnya resolusi yang digunakan adalah 320 x 240 dengan terlebih dahulu melalui konversi citra RGB to grayscale. *Over spatter* adalah percikan las yang terlalu banyak memenuhi bidang las. *Over spatter* dapat disebabkan karena arus terlalu besar, busur las terlalu jauh, dan electrode meyerap banyak uap. *Over spatter* dapat menurunkan kerapian dalam pengelasan dan dapat menimbulkan terjadinya *prosity*. bentuk cacat las *over spatter*

ini sangat mudah dikenali yaitu terdapat bintik-bintik menonjol disekitar bidang las akibat percikan material las yang terlalu berlebihan. Dari hasil pengujian yang dilakukan untuk hasil pengelasan yang mempunyai cacat *over spatter*, arsitektur CNN yang digunakan sudah dapat mengenali cacat over spatter sejumlah 17 data dari 21 data. beberapa hasil klasifikasi yang ditunjukkan pada Tabel 4.13.









Tabel 4.13 Hasil Klasifikasi Pengelasan dengan Cacat *Over Spatter*.

No	Hasil Klasifikasi	Keterangan
1		Benar
2		Benar
3		Benar
4		Benar
5		Benar
6		Benar
7		Salah
8		Salah
9		Salah

Uji coba yang ke tiga bertujuan untuk mengklasifikasikan hasil pengelasan yang mempunyai cacat *porosity*, *porosity* merupakan sekelompok gelembung gas

yang terjebak di dalam material las. *Porosity* bisa terjadi karena proses pemadatan yang terlalu cepat. *Porosity* berupa rongga-rongga kecil berbentuk bola yang mengelompok pada lokasi-lokasi pengelasan. Terkadang terjadi rongga besar berbentuk bola tunggal atau tidak mengelompok. Untuk kategori cacat ini jumlah data yang dihasilkan terdiri 55 citra, sama halnya dengan pengujian sebelumnya resolusi yang digunakan adalah 320 x 240 dengan terlebih dahulu melalui konversi citra RGB to grayscale. Dari hasil pengujian yang dilakukan untuk hasil pengelasan yang mempunyai cacat *porosity*, arsitektur CNN yang digunakan sudah dapat mengenali hasil pengelasan yang mempunyai cacat *porosity* sejumlah 44 data dari total 55 data, beberapa hasil klasifikasi seperti yang ditunjukkan Tabel 4.14.






Tabel 4.14 Hasil Klasifikasi Pengelasan dengan Cacat *Porosity*.




No	Hasil Klasifikasi			Keterangan
1				Benar
2				Benar
3				Benar
4				Benar
5				Salah
6				Benar
7				Salah
8				Salah

Uji coba yang terakhir bertujuan untuk mengklasifikasikan hasil pengelasan yang mempunyai cacat *Undercut*, *Undercut* merupakan istilah yang digunakan untuk

menggambarkan sebuah alur (*groove*) benda kerja yang mencair dan terletak pada tepi atau kaki pengelasan (manik-manik las) dimana alur benda kerja yang mencair tersebut tidak terisi oleh cairan las. *Undercut* disebabkan oleh arus yang berlebihan dalam proses pengelasan. Untuk kategori cacat ini jumlah data yang dihasilkan terdiri 33 citra, sama halnya dengan pengujian sebelumnya resolusi yang digunakan adalah 320 x 240 dengan terlebih dahulu melalui konversi citra RGB to grayscale. Dari hasil pengujian yang dilakukan untuk hasil pengelasan yang mempunyai cacat undercut arsitektur CNN yang digunakan sudah dapat mengenali hasil pengelasan yang baik sejumlah 30 data dari total 3 data, beberapa hasil klasifikasi seperti yang ditunjukkan pada Tabel 4.15. Dari empat percobaan yang telah dilakukan secara *real time* terhadap hasil pengelasan didapatkan rata-rata keberhasilan adalah 77.75% sesuai dengan data yang ditunjukkan pada Tabel 4.16.

.Tabel 4.15 Hasil Klasifikasi Pengelasan dengan Cacat *Undercut*

No	Hasil Klasifikasi	Keterangan
1		Benar
2		Benar
3		Benar
4		Benar
5		Benar
6		Benar

No	Hasil Klasifikasi	Keterangan
7		Salah
8		Salah
9		Benar

Tabel 4.16 Persentase Keberhasilan Pengujian

No	Kategori Percobaan	Total	Hasil Klasifikasi	
			Benar	Salah
1	<i>Good</i>	108	86	21
2	<i>Over Spatter</i>	21	17	4
3	<i>Porosity</i>	55	44	11
4	<i>Undercut</i>	33	30	33
Total Data		217	177	69
Persentase Keberhasilan		81 %		

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Pada penelitian ini telah didapatkan struktur CNN yang tepat yaitu arsitektur CNN 2, dengan rincian sebagai berikut; matriks kernel untuk proses konvolusi berukuran 5 x 5 menggunakan kernel Gaussian dengan jumlah filter konvolusi yaitu 10 dan matriks untuk proses subsampling berukuran 2 x 2. Jumlah vektor input untuk Neural Network adalah 168.400 dan vektor output adalah 4. Arsitektur jaringan syaraf tiruan memiliki 1 layer tersembunyi (single hidden layer) dengan 64 hidden Neuron. Jumlah iterasi 60 dan laju pembelajaran α yaitu 0,2. Arsitektur yang telah disebutkan pada poin diatas mampu menghasilkan akurasi untuk klasifikasi citra las sebesar 70,85 persen dari total 24 data yang diujikan, atau CNN pada penelitian ini mampu mengklasifikasikan 17 data citra dengan benar. *Feature maps* atau banyaknya fitur konvolusi yang digunakan, perubahan nilai laju pembelajaran dan jumlah Neuron mempengaruhi akurasi yang dihasilkan. Sedangkan pengujian klasifikasi yang dilakukan secara *real time* mempunyai tingkat keberhasilan 81%.

5.2 Saran

1. Variasi jumlah dan jenis kernel perlu dimodifikasi untuk mengoptimalkan akurasi pada penelitian selanjutnya.
2. Perlu dilakukan penelitian lebih lanjut tentang korelasi laju pembelajaran, filter konvolusi, jumlah hidden Neuron untuk mengoptimalkan hasil akurasi data validasi.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- Andreas, 2015. *Controlling The Hidden Layers' Output To Optimizing The Training Process in The Deep Neural Network Algorithm*. Shenyang, China, s.n.
- Fukushima, K., 1980. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, Volume 36, pp. 193-202.
- G. E. Hinton, S. O. a. Y.-W. T., 2006. A Fast Learning Algorithm. *Neural Computation*, 18(7), pp. 1527-1554.
- I Wayan Suartika E. P, A. Y. W. d. R. S., 2016. Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101. *JURNAL TEKNIK ITS*, Volume 5.
- J. Mirapeix, 2007. Real-time arc-welding defect detection and classification with principal. *NDT&E International*, Volume 40, pp. 315 - 325.
- J. T. Springenberg, 2015. Striving For Simplicity: The All Convolutional Net. *ICLR 2015*.
- Jayendra Kumar, R. A. S. S., 2014. *Multi - Class Welding Flaws Classification using Texture Feature for Radiographic Images*. Roorkee, India: Advances in Electrical Engineering (ICAEE), 2014 International Conference on.
- Jurandir Primo, P., 2012. *Welding Inspection Qualifications & Testing Procedures*. s.l.:PDHonline Course M415 .
- Karen Simonyan, A. Z., 2014. *Two-Stream Convolutional Networks for Action Recognition in Videos*. s.l., Visual Geometry Group, University of Oxford.
- Karpathy, A., 2014. *Large-scale Video Classification with Convolutional Neural Networks*. Computer Science Department, Stanford University, CVPR.
- Mauridhi Hery, A. K., 2006. *Supervised Neural Networks dan Aplikasinya*. Yogyakarta: Graha Ilmu.
- Norvig, S. J. R. a. P., 1995. *Artificial Intelligence A Modern Approach*.
- Stanford, 2016. *An Introduction to Convolutional Neural Network*. [Online] Available at: <http://scarlet.stanford.edu>. [Accessed 23 11 2016].
- Stathakis, D., 2008. How Many Hidden Layers And Nodes?. *International Journal of Remote Sensing*.
- Y.Le Cun, B. B., 1990. *Handwritten Digit Recognition with a Back Propagation Network*. s.l.:s.n.

Halaman ini sengaja dikosongkan

BIODATA PENULIS



I. Data Pribadi

Nama : Agus Khumaidi
Tempat/Tanggal Lahir : Gresik, 17 Agustus 1993
Alamat : Jl. Dewi Sekardadu No.09 RT.01 RW.02 Ngargosari
Gresik – Jawa Timur
Agama : Islam
Email : agus15@mhs.ee.its.ac.id

II. Riwayat Pendidikan

Pendidikan Dasar dan Menengah

Jenjang	Nama Sekolah	Tahun Lulus
SD	SD Muhammadiyah 1 Giri	2005
SMP	SMP Muhammadiyah 4 Giri	2008
SMA	SMK Negeri 1 Cerme	2011

Pendidikan Tinggi

Perguruan Tinggi	Tahun Lulus	Jurusan	Gelar
Politeknik Perkapalan Negeri Surabaya	2015	Teknik Otomasi	S.ST

Halaman ini sengaja dikosongkan